

# Interstage Information Integrator V10.3.0



## システム設計ガイド

Windows/Solaris/Linux

B1WD-3013-04Z0(00)  
2012年2月

# まえがき

---

本書は、以下の製品を対象としています。

- Interstage Information Integrator V10.3.0

以下の各オプション製品についても対象としています。

- Interstage Information Integratorファイル転送エージェント V10.3.0
- Interstage Information Integrator統合管理オプション V10.3.0

## 本書の目的

本書は、Interstage Information Integrator(以降、Information Integratorと略します)の機能の概要、設計方法、およびシステム構成について説明しています。

## 本書の読者

本書は、以下の読者を対象としています。

- Information Integratorの運用環境を設計する人

本書を読む場合、以下の知識が必要です。

- OSに関する一般的な知識
- データベースの機能、使用方法に関する一般的な知識
- パブリック・クラウドに関する一般的な知識
- 帳票に関する一般的な知識

## 本書の構成

### 第1章 Information Integratorの機能

Information Integratorの機能概要、および各機能について説明しています。

### 第2章 Information Integratorの導入から運用までの流れ

Information Integratorの導入から運用までの作業の流れを説明しています。

### 第3章 ユーザー設計

Information Integratorを利用するユーザーの種類、その役割について説明しています。

### 第4章 システム構成

Information Integratorで利用する各コンポーネントの説明や、データソースの種類、または運用に合わせたシステム構成について説明しています。

### 第5章 処理プロセスの設計

Information Integratorで行う、収集、変換、配付などの処理プロセスの設計方法について説明しています。

### 第6章 III Studioを利用した設計

III Studioを利用した処理プロセスの設計手順について説明します。

### 第7章 デザインシートを利用した設計

デザインシートを利用した処理プロセスの設計手順について説明します。

### 第8章 動作環境の設計

Information Integratorを運用するための動作環境の設計方法について説明しています。

### 第9章 セキュリティ設計

Information Integratorを運用するためのセキュリティの設計方法について説明しています。

## 第10章 文字コード変換の準備

文字コード変換機能を利用する場合に、標準機能以外の文字コードを変換する場合の設定方法について説明しています。

### 付録A 各種の利用可能機能一覧

通信プロトコルごと、抽出・格納対象のデータベースごとの利用可能機能について一覧で表しています。

### 付録B Information Integratorで扱えるデータ形式について

Information Integratorで扱えるデータ形式の詳細な条件について説明しています。

### 付録C ファイル転送エージェントで利用可能な機能一覧

ファイル転送エージェントで利用可能な機能について説明しています。

### 付録D フォーマット形式について

Information Integratorで対応するファイル形式のフォーマットについて説明しています。

### 付録E Information IntegratorがサポートするFTPのコマンド

Information IntegratorがサポートするFTPコマンドを説明しています。

### 付録F 変換・編集時に行うデータチェックについて

変換・編集機能を利用するときに行うデータチェックの内容について説明しています。

### 付録G ファイアウォール環境への導入時の注意事項

Information Integratorをファイアウォール環境へ導入する場合の注意事項を説明しています。

### 付録H 組み込み可能なアプリケーション

プラグイン機能を利用する場合に、組み込み可能なアプリケーションについて説明しています。

### 付録I プラグイン機能を利用した帳票出力

プラグイン機能を利用して帳票出力を行う方法について説明しています。

### 付録J サービスバス連携を行う場合のService Integratorの設定

サービスバス連携入力または、サービスバス連携出力を行う場合のService Integratorの設定について説明しています。

### 付録K プラグイン機能を利用したExcel連携

プラグイン機能を利用してExcel連携を行う方法について説明しています。

## 本書の位置付け

### Interstage Information Integrator マニュアル体系と読み方

Information Integratorのマニュアル体系と、利用者の用途・目的に合わせた参照マニュアルについて記載しています。

### Interstage Information Integrator リリース情報

Information Integratorについて、バージョンアップした機能の概要と内容、詳細(マニュアルの該当箇所)、および互換に関する情報を記載しています。

### Interstage Information Integrator システム設計ガイド[本書]

Information Integratorについて、機能の概要、設計およびシステム構成について記載しています。

### Interstage Information Integrator セットアップガイド

Information Integratorのインストール後の環境構築方法について記載しています。

### Interstage Information Integrator 運用ガイド

Information Integratorの運用方法について記載しています。

### Interstage Information Integrator コマンドリファレンス

Information Integratorが提供するコマンドについて記載しています。

### Interstage Information Integrator メッセージ集

Information Integratorのメッセージについて記載しています。

## Interstage Information Integrator 用語集

Information Integratorのマニュアルで使用する用語について記載しています。

### 本書の表記について

#### マニュアル名称

本書に記載されているマニュアルの名称は、以下のように省略して表記します。

略称	正式名称
マニュアル体系と読み方	Interstage Information Integrator マニュアル体系と読み方
リリース情報	Interstage Information Integrator リリース情報
システム設計ガイド(本書)	Interstage Information Integrator システム設計ガイド
セットアップガイド	Interstage Information Integrator セットアップガイド
運用ガイド	Interstage Information Integrator 運用ガイド
コマンドリファレンス	Interstage Information Integrator コマンドリファレンス
メッセージ集	Interstage Information Integrator メッセージ集
用語集	Interstage Information Integrator 用語集

#### 注意事項

本書の“1.4.3 項目編集”、“1.4.4 レコード編集”で記載している【例】の図は表示例です。わかりやすくするため項目見出しを付けていますが、実際の製品では、項目見出しは利用できません。

#### 略称

本書に記載されている製品の名称は、以下のように省略して表記します。

なお、本書では、システム名または製品名に付記される登録表示(TM)または®は、省略しています。

略称	正式名称
Windows	Microsoft Windows Server 2008 Foundation、 Microsoft Windows Server 2008 Standard、 Microsoft Windows Server 2008 Enterprise、 Microsoft Windows Server 2008 Datacenter、 Microsoft Windows Server 2008 Standard without Hyper-V、 Microsoft Windows Server 2008 Enterprise without Hyper-V、 Microsoft Windows Server 2008 Datacenter without Hyper-V、 Microsoft Windows Server 2008 R2 Foundation、 Microsoft Windows Server 2008 R2 Standard、 Microsoft Windows Server 2008 R2 Enterprise、 Microsoft Windows Server 2008 R2 Datacenter、 Microsoft Windows Server 2003, Standard Edition、 Microsoft Windows Server 2003, Enterprise Edition、 Microsoft Windows Server 2003, Datacenter Edition、 Microsoft Windows Server 2003, Standard x64 Edition、 Microsoft Windows Server 2003, Enterprise x64 Edition、 Microsoft Windows Server 2003, Datacenter x64 Edition、 Microsoft Windows Server 2003 R2, Standard Edition、 Microsoft Windows Server 2003 R2, Enterprise Edition、 Microsoft Windows Server 2003 R2, Datacenter Edition、 Microsoft Windows Server 2003 R2, Standard x64 Edition、 Microsoft Windows Server 2003 R2, Enterprise x64 Edition、 Microsoft Windows Server 2003 R2, Datacenter x64 Edition、 Windows®7 Home Premium、 Windows®7 Professional、

略称	正式名称
	Windows®7 Enterprise、 Windows®7 Ultimate、 Windows® Vista Home Basic、 Windows® Vista Home Premium、 Windows® Vista Business、 Windows® Vista Ultimate、 Windows® Vista Enterprise、 Windows® XP Home Edition、 Windows® XP Professional および Windows® XP
Windows XP	Windows® XP Home Edition および Windows® XP Professional
Windows Vista	Windows® Vista Home Basic、 Windows® Vista Home Premium、 Windows® Vista Business、 Windows® Vista Ultimate および Windows® Vista Enterprise
Windows 7	Windows®7 Home Premium、 Windows®7 Professional、 Windows®7 Enterprise および Windows®7 Ultimate
Windows Server 2003	Microsoft Windows Server 2003, Standard Edition、 Microsoft Windows Server 2003, Enterprise Edition、 Microsoft Windows Server 2003, Datacenter Edition、 Microsoft Windows Server 2003, Standard x64 Edition、 Microsoft Windows Server 2003, Enterprise x64 Edition、 Microsoft Windows Server 2003, Datacenter x64 Edition、 Microsoft Windows Server 2003 R2, Standard Edition、 Microsoft Windows Server 2003 R2, Enterprise Edition、 Microsoft Windows Server 2003 R2, Datacenter Edition、 Microsoft Windows Server 2003 R2, Standard x64 Edition、 Microsoft Windows Server 2003 R2, Enterprise x64 Edition および Microsoft Windows Server 2003 R2, Datacenter x64 Edition
Windows Server 2008	Microsoft Windows Server 2008 Foundation、 Microsoft Windows Server 2008 Standard、 Microsoft Windows Server 2008 Enterprise、 Microsoft Windows Server 2008 Datacenter、 Microsoft Windows Server 2008 Standard without Hyper-V、 Microsoft Windows Server 2008 Enterprise without Hyper-V、 Microsoft Windows Server 2008 Datacenter without Hyper-V、 Microsoft Windows Server 2008 R2 Foundation、 Microsoft Windows Server 2008 R2 Standard、 Microsoft Windows Server 2008 R2 Enterprise および Microsoft Windows Server 2008 R2 Datacenter
フェールオーバー クラスタ	Windows Server 2008 フェールオーバー クラスタ および Microsoft Cluster Service
Excel	Microsoft® Excel
Word	Microsoft® Word
SQL Server	Microsoft® SQL Server™
Oracle	Oracle® Enterprise Edition、 Oracle® Database Standard Edition、

略称	正式名称
	Oracle® Database Standard Edition One および Oracle® Database Enterprise Edition
Solaris	Oracle Solaris
Linux	Red Hat Enterprise Linux
UNIX	Solaris および Linux
XL-DATA/MV	XLデータムーバ、 Softtek XLデータムーバ、 ETERNUS SF XL-DATA/MV、 OSIV MSP XL-DATA/MV および OSIV MSP XL-DATA/MV
Interstage Application Server	Interstage® Application Server Standard-J Edition および Interstage® Application Server Enterprise Edition
Shunsaku	Interstage® Shunsaku Data Manager Enterprise Edition
Data Effector	Interstage® Data Effector Standard Edition
Charset Manager	Interstage® Charset Manager
Linkexpress	Linkexpress、 Linkexpress Advanced Edition、 Linkexpress Standard Edition および Linkexpress Enterprise Edition
Symfoware Server	Symfoware® Server Standard Edition、 Symfoware® Server Enterprise Edition および Symfoware® Server Enterprise Extended Edition
List Creator	Interstage® List Creator Enterprise Edition および Interstage® List Creator Standard Edition
DB2	IBM® DB2 9.7 for Linux, UNIX and Windows, Express Edition IBM® DB2 9.7 for Linux, UNIX and Windows, Workgroup Server Edition および IBM® DB2 9.7 for Linux, UNIX and Windows, Enterprise Server Edition
Information Storage	Interstage® Information Storage
Service Integrator	Interstage® Service Integrator

## 輸出管理規制について

本ドキュメントを輸出または提供する場合は、外国為替、外国貿易法および米国輸出管理関連法規等の規制をご確認の上、必要な手続きをおとりください。

## 商標

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

UNIXは、米国ならびに他の国におけるオープン・グループの登録商標です。

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

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

Red Hat、RPMおよびRed Hatをベースとしたすべての商標とロゴは、Red Hat, Inc.の米国およびその他の国における商標または登録商標です。

HULFTは、株式会社セゾン情報システムズの登録商標です。

Salesforceは、株式会社セールスフォース・ドットCOMの登録商標です。

DB2は、世界の多くの国で登録されたInternational Business Machines Corp.の商標です。

PostgreSQLは、PostgreSQLの米国およびその他の国における商標です。

Interstage、Linkexpress、Symfoware、Shunsaku、Systemwalker、List Creator、Information StorageおよびService Integratorは、富士通株式会社の登録商標です。

そのほか、本書に記載されている商標および登録商標は、各社の商標または登録商標です。

#### **出版年月および版数**

平成24年2月

平成24年 2月 初版

#### **著作権表示**

- 本書を無断で他に転載しないようお願いします。
- 本書は予告なしに変更されることがあります。

Copyright FUJITSU LIMITED 2012

# 目次

第1章 Information Integratorの機能	1
1.1 機能概要	1
1.2 データ収集	1
1.2.1 Information Integratorで収集可能なデータ形式	2
1.2.2 データ収集方法	2
1.2.3 ファイル受信	3
1.2.3.1 ファイル受信のオプション機能	5
1.2.4 ファイル入力	5
1.2.5 DB抽出	6
1.2.6 オブジェクト抽出	7
1.2.7 IIS抽出	7
1.2.8 BLOB収集	8
1.2.9 サービスバス連携入力	8
1.2.10 Excel連携入力	9
1.3 データ配付	9
1.3.1 Information Integratorで配付可能なデータ形式	9
1.3.2 データの配付方法	11
1.3.3 ファイル送信	11
1.3.3.1 ファイル送信のオプション機能	13
1.3.4 ファイル出力	14
1.3.5 DB格納	14
1.3.6 オブジェクト格納	15
1.3.6.1 オブジェクト格納のオプション機能	15
1.3.7 帳票出力	15
1.3.8 IIS格納	16
1.3.9 BLOB配付	16
1.3.9.1 BLOB配付のオプション機能	17
1.3.10 サービスバス連携出力	17
1.3.11 クラウド連携転送データ分割機能	18
1.3.12 Excel連携出力	18
1.4 データ変換・編集	18
1.4.1 ファイル形式変換	19
1.4.2 文字コード変換	20
1.4.3 項目編集	22
1.4.3.1 項目並び替え	22
1.4.3.2 項目選択	22
1.4.3.3 項目追加	23
1.4.3.4 属性変換	23
1.4.3.5 トリム	32
1.4.3.6 パディング	33
1.4.3.7 文字列の部分抽出	34
1.4.3.8 文字列の設定	35
1.4.3.9 文字列の連結	35
1.4.3.10 大文字/小文字変換	36
1.4.3.11 全角/半角変換	36
1.4.3.12 ひらがな/カタカナ変換	37
1.4.3.13 濁点/半濁点変換	37
1.4.3.14 四則演算	37
1.4.3.15 項目条件判定	38
1.4.3.16 数字表記変換	38
1.4.3.17 日時の書式変換	39
1.4.3.18 日時の部分取得	40
1.4.3.19 項目編集時の注意事項	41
1.4.4 レコード編集	41



1.4.4.1 振分け	41
1.4.4.2 ソート	43
1.4.4.3 連結(ジョイン、ユニオン)	43
1.4.4.4 集計	46
1.5 相手側ジョブ起動	47
1.5.1 相手側ジョブ起動	47
1.5.2 ジョブ結果通知の有無	49
1.5.3 ジョブの起動インタフェース	50
1.5.4 相手側ジョブ起動の動作環境	51
1.5.5 ジョブの起動パラメタ	52
1.5.6 ジョブがInformation Integratorに行う情報通知	53
1.5.7 ジョブのサンプル	54
1.6 プロセス制御機能	54
1.6.1 処理プロセスの実行	54
1.6.2 自動リトライ	55
1.6.3 排他制御	55
1.6.4 順序性制御	56
1.6.5 イベント監視	56
1.6.5.1 ディレクトリを監視する場合	58
1.6.5.2 複数のイベントを監視する場合	59
1.6.5.3 イベントの監視に休止期間を設ける・一時停止する	61
1.7 プラグイン	62
1.7.1 処理プラグイン機能のイメージ	62
1.7.2 変換プラグイン機能のイメージ	63
1.8 管理機能	64
1.8.1 定義作成/定義管理	64
1.8.1.1 メタ収集機能	65
1.8.2 データ管理	66
1.8.3 運用管理	69
1.9 ファイル転送エージェント機能	70
1.9.1 Information Integratorサーバで一元管理する	71
1.9.2 処理プロセスの運用方法	71
1.10 SSL通信を利用する	73
1.10.1 証明書管理について	73
1.10.2 SSL通信のサポート範囲	74
<b>第2章 Information Integratorの導入から運用までの流れ</b>	<b>76</b>
2.1 運用までの流れ	76
<b>第3章 ユーザー設計</b>	<b>78</b>
3.1 ユーザーの種類	78
3.2 各ユーザーの操作範囲	79
3.3 処理プロセスの操作・運用を行う運用ユーザーの設計	80
<b>第4章 システム構成</b>	<b>82</b>
4.1 システム構成	82
4.1.1 Information Integratorの構成物	82
4.1.2 標準構成	83
4.1.3 負荷分散構成	84
4.1.4 Information Integrator統合管理構成	85
4.1.5 クラスタ構成	85
4.2 相手側システムの接続について	86
4.2.1 ファイル送受信の接続について	86
4.2.1.1 ファイル送受信で扱える通信プロトコル	86
4.2.1.2 Information Integratorが接続可能なデータ集配信製品	87
4.2.1.3 ファイルシステムの留意事項について	88
4.2.2 オブジェクト抽出・格納時の通信プロトコル	89
4.2.3 BLOB収集・配付時の通信プロトコル	89

4.3 相手側システムの種類ごとのシステム構成.....	89
4.3.1 相手側システムがWindows、Solaris、Linuxの場合.....	89
4.3.2 相手側システムがグローバルサーバ/PRIMEFORCEの場合.....	91
4.3.3 相手側システムがASPの場合.....	92
4.3.4 HULFTを利用する場合.....	93
4.3.5 標準FTPツールを利用する場合.....	93
4.3.6 相手側システムがSalesforceの場合.....	94
4.3.7 帳票データを出力する場合.....	95
4.3.8 相手側システムがInformation Storageの場合.....	96
4.3.9 相手側システムがAzure Storageの場合.....	96
4.4 ファイアウォールを利用する場合.....	97
4.5 Internet Protocolについて.....	97
4.6 DNSを利用する場合.....	98
4.7 相手側システム上にInformation Integratorサーバを導入する場合.....	99
<b>第5章 処理プロセスの設計.....</b>	<b>101</b>
5.1 処理プロセスの設計の概要.....	101
5.1.1 Information Integratorの処理プロセス.....	101
5.1.2 処理プロセスの設計の流れ.....	101
5.2 処理プロセスの設計作業概要.....	103
<b>第6章 III Studioを利用した設計.....</b>	<b>104</b>
6.1 III Studioの起動・停止.....	104
6.1.1 III Studioの起動.....	104
6.1.2 III Studioの終了.....	108
6.2 III Studioの画面構成.....	108
6.3 III Studioの利用方法.....	109
6.3.1 定義の新規作成.....	109
6.3.2 定義をサーバに登録.....	111
6.3.3 サーバに登録した定義の修正方法.....	111
6.3.4 定義の一時保存・再編集.....	112
6.3.5 定義の複写.....	112
6.4 処理プロセスの処理手順の設計.....	112
6.4.1 ファンクション処理間のデータ受け渡しの設計.....	116
6.5 処理プロセスのパラメタの設計.....	117
6.5.1 処理プロセスのパラメタを設計する定義種別.....	117
6.5.2 処理プロセスのパラメタの設計手順.....	118
6.5.3 見出し行の存在するデータを扱う場合の留意事項.....	118
6.6 データ収集処理の設計.....	119
6.6.1 収集元システムの設計.....	119
6.6.2 収集データの設計.....	120
6.6.3 収集方法の設計.....	121
6.7 データ配付処理の設計.....	123
6.7.1 データ配付先システムの設計.....	123
6.7.2 配付データの設計.....	124
6.7.3 配付方法の設計.....	125
6.8 処理データの構造・フォーマットの設計.....	126
6.8.1 処理データの設計概要.....	126
6.8.2 操作手順.....	127
6.8.3 データ構造の設計.....	127
6.8.4 メタ収集機能を利用した設計手順.....	128
6.8.5 データフォーマットの設計.....	130
6.9 データ変換処理の設計.....	130
6.9.1 データ変換処理の設計.....	131
6.9.1.1 データ変換定義の作成.....	131
6.9.1.2 データ構造の読み込み.....	131
6.9.1.3 データフォーマットの設定.....	133
6.9.1.4 項目対応付けの設計手順.....	133

6.9.1.5	項目編集処理の設計手順	134
6.9.1.6	ソート処理の設計手順	136
6.9.1.7	集計処理の設計手順	136
6.9.1.8	変換処理の詳細設定の設計手順	137
6.9.1.9	結合処理の設計手順	137
6.9.2	データ振分処理の設計	138
6.9.2.1	データ振分定義の作成	138
6.9.2.2	データ振分処理の設計	138
6.10	プラグイン処理の設計	138
6.10.1	処理プラグイン機能の設計手順	138
6.10.2	変換プラグイン機能の設計手順	139
6.10.3	データを圧縮または解凍する場合	140
6.10.3.1	ificompcmd(データ圧縮コマンド)	140
6.10.3.2	ifiuncompcmd(データ解凍コマンド)	141
6.10.4	ファイル転送エージェントの処理プロセスと連携する場合(オプション機能)	142
6.10.4.1	ificlexprc(エージェント処理プロセス連携用コマンド)	142
<b>第7章</b>	<b>デザインシートを利用した設計</b>	<b>144</b>
7.1	デザインシートの利用方法	144
7.1.1	デザインシートの新規作成	144
7.1.2	デザインシートの入力内容のチェック	145
7.1.3	デザインシートをサーバに登録	146
7.1.4	サーバに登録した定義内容の修正方法	147
7.1.5	デザインシートの一時保存	148
7.1.6	デザインシートの複写	149
7.2	処理プロセスの処理手順の設計	149
7.2.1	ファンクション処理間のデータ受け渡しの設計	151
7.3	処理プロセスのパラメタの設計	152
7.3.1	デザインシートの種類	152
7.3.2	処理プロセスのパラメタの設計手順	153
7.3.3	見出し行の存在するデータを扱う場合の留意事項	154
7.4	データ収集処理の設計	154
7.4.1	収集元システムの設計	154
7.4.2	収集データの設計	155
7.4.3	収集方法の設計	155
7.5	データ配付処理の設計	158
7.5.1	データ配付先システムの設計	158
7.5.2	配付データの設計	159
7.5.3	配付方法の設計	159
7.6	処理データの構造・フォーマットの設計	161
7.6.1	処理データの設計概要	161
7.6.2	操作手順	162
7.6.3	データ構造の設計	162
7.6.4	メタ収集機能を利用した設計手順	163
7.6.5	データフォーマットの設計	165
7.7	データ変換処理の設計	165
7.7.1	データ変換処理の設計	166
7.7.1.1	データ変換定義シートの作成	166
7.7.1.2	データ構造の読み込み	166
7.7.1.3	データフォーマットの設定	169
7.7.1.4	項目対応付けの設計手順	169
7.7.1.5	項目編集処理の設計手順	169
7.7.1.6	結合処理の設計手順	171
7.7.1.7	ソート処理の設計手順	172
7.7.1.8	集計処理の設計手順	173
7.7.1.9	変換処理の詳細設定の設計手順	173
7.7.2	データ振分処理の設計	173

7.7.2.1 データ振分定義シートの作成	173
7.7.2.2 データ振分処理の設計	173
7.8 プラグイン処理の設計	174
7.8.1 処理プラグイン機能の設計手順	174
7.8.2 変換プラグイン機能の設計手順	175
7.8.3 データを圧縮または解凍する場合	175
7.8.4 ファイル転送エージェントの処理プロセスと連携する場合	176
<b>第8章 動作環境の設計</b>	<b>177</b>
8.1 多重度の設計について	177
8.1.1 プロセス受付多重度の設計について	177
8.1.2 プロセス実行多重度の設計について	177
8.1.3 データ収集/データ配付/プラグインの各ファンクションの実行多重度の設計について	177
8.1.4 データ変換ファンクションの実行多重度の設計について	178
8.1.5 ファイル送受信多重度の設計について	178
8.1.5.1 回線の同時使用数の調整	178
8.1.5.2 多重度の見積り例(相手側ジョブ起動)	179
8.1.5.3 多重度の見積り例(起動側ファイル送信)	180
8.1.5.4 多重度の見積り例(起動側ファイル受信)	181
8.1.5.5 多重度の見積り例(ファイル転送エージェントと連携する場合)	181
8.2 データベースの環境準備	182
8.2.1 各データベースに共通の環境設定	183
8.2.2 Symfoware Serverの環境設定	184
8.2.3 Oracleの環境設定	186
8.2.4 SQL Serverの環境設定	187
8.2.5 DB2の環境設定	188
8.2.6 PostgreSQLの環境設定	188
8.2.7 SQL Azureの環境設定	189
8.2.8 その他のデータベース(ODBC接続)の環境設定	190
8.2.9 ODBCの環境設定について(Windowsの場合)	190
8.2.10 ODBCの環境設定について(UNIXの場合)	190
8.2.10.1 ODBC設定ファイルの編集	190
8.2.10.2 ODBC設定ファイルの記述形式	191
8.2.10.3 ODBC設定ファイルのセクション名、キー名、パラメタ値	191
8.2.10.4 ODBC設定ファイルの記述例	193
8.2.11 留意事項	193
8.2.11.1 データベースに共通の留意点	193
8.2.11.2 Symfoware Serverを使用する上での留意点	194
8.2.11.3 Oracleを使用する上での留意点	195
8.2.11.4 SQL Serverを使用する上での留意点	196
8.2.11.5 DB2を使用する上での留意点	197
8.2.11.6 PostgreSQLを使用する上での留意点	197
8.2.11.7 その他のデータベース(ODBC接続)を使用する上での留意点	197
8.3 Interstage List Creatorの環境準備	198
<b>第9章 セキュリティ設計</b>	<b>199</b>
9.1 利用者の権限	199
9.2 ファイル送受信、相手側ジョブ起動利用時のセキュリティ設定情報	199
9.2.1 利用者認証(ログイン検査)	199
9.2.2 相手側ジョブ起動の実行資格	199
9.2.3 接続システムごとに要求されるセキュリティ情報	199
9.3 DB抽出・DB格納利用時のセキュリティ設定情報	200
9.4 オブジェクト抽出・格納利用時のセキュリティ設定情報	201
9.5 帳票出力利用時のセキュリティ設定情報	201
9.6 BLOB収集・配付利用時のセキュリティ設定情報	202
9.7 Windowsでのセキュリティ注意事項	202
<b>第10章 文字コード変換の準備</b>	<b>204</b>

10.1 文字コード変換機能の環境設定.....	204
10.1.1 エラー発生時の動作.....	204
10.1.2 文字コード変換テーブルの切り替え.....	205
10.1.3 文字コード変換テーブルのカスタマイズ.....	206
10.1.3.1 カスタマイズ方法.....	207
10.2 制御ファイルを使用したカスタマイズ例.....	208
10.3 Charset Manager文字コード変換表の取り込み例.....	210
<b>付録A 各種の利用可能機能一覧.....</b>	<b>211</b>
A.1 通信プロトコルごとの利用可能機能一覧.....	211
A.2 データベースごとの利用可能機能一覧.....	212
A.3 オブジェクトごとの利用可能機能一覧.....	214
<b>付録B Information Integratorで扱えるデータ形式について.....</b>	<b>215</b>
B.1 収集できるデータ形式.....	215
B.1.1 Information Integratorで収集できるCSVファイル形式.....	216
B.1.2 Information Integratorで収集できるバイナリファイル形式.....	219
B.1.3 Information Integratorで収集できるXMLファイル形式.....	220
B.1.4 Information Integratorで収集できるデータベース形式.....	223
B.1.5 Information Integratorで収集できるオブジェクト形式.....	235
B.1.6 Information Integratorで収集できるInformation Storage形式.....	238
B.1.7 Information Integratorで収集できるBLOB形式.....	239
B.2 配付できるデータ形式.....	239
B.2.1 Information Integratorで配付できるCSVファイル形式.....	240
B.2.2 Information Integratorで配付できるバイナリファイル形式.....	240
B.2.3 Information Integratorで配付できるXMLファイル形式.....	240
B.2.4 Information Integratorで配付できるデータベース形式.....	241
B.2.5 Information Integratorで配付できるオブジェクト形式.....	247
B.2.6 Information Integratorで出力できる帳票データ.....	249
B.2.7 Information Integratorで配付できるInformation Storage形式.....	255
B.2.8 Information Integratorで配付できるBLOB形式.....	256
B.3 Information Integratorで対応するデータ型のデータ例.....	256
B.4 変換処理できる項目属性の一覧.....	257
B.5 CSVファイルで扱えるデータ属性.....	261
B.6 バイナリファイルで扱えるデータ属性.....	264
B.7 XMLファイルで扱えるデータ属性.....	264
B.8 列属性について.....	264
<b>付録C ファイル転送エージェントで利用可能な機能一覧.....</b>	<b>267</b>
<b>付録D フォーマット形式.....</b>	<b>269</b>
D.1 一般フォーマット.....	269
D.2 一般フォーマット(繰返しあり).....	270
D.3 レコード終端可変長形式フォーマット.....	271
D.4 繰返し数指定可変長フォーマット.....	271
D.5 伝票形式フォーマット.....	272
D.6 複数フォーマットが混在するフォーマット.....	275
<b>付録E Information IntegratorがサポートするFTPのコマンド.....</b>	<b>279</b>
<b>付録F 変換・編集時に行うデータチェックについて.....</b>	<b>281</b>
F.1 データ形式に対するチェック項目.....	281
F.1.1 全角文字列に対するデータチェック.....	281
F.1.2 日付・時間属性に対するデータチェック.....	281
F.1.3 数値属性に対するデータチェック.....	282
F.1.4 バイナリ属性に対するデータチェック.....	283
F.1.5 ビット属性に対するデータチェック.....	284
F.2 桁数の過不足に対するデータチェック.....	284
F.3 必須入力データのチェック.....	285

F.3.1 NULL値のチェック.....	285
付録G ファイアウォール環境への導入時の注意事項.....	286
G.1 FTP/FTP+/FTPMでのコネクション確立シーケンス(前提知識).....	286
G.2 ファイアウォール環境での問題発生箇所と対処.....	287
G.2.1 フィルタリング.....	287
G.2.2 NAT変換.....	288
付録H 組み込み可能なアプリケーション.....	290
H.1 処理プラグイン機能で組み込み可能なアプリケーション.....	290
H.2 変換プラグイン機能で組み込み可能なアプリケーション.....	292
H.2.1 アプリケーションの開発環境.....	292
H.2.2 アプリケーションの利用権限.....	293
H.2.3 利用可能なアプリケーションインタフェース.....	293
H.2.4 アプリケーションで利用可能なデータの文字コード.....	294
H.2.5 利用可能な引数の型.....	294
H.2.6 利用可能なデータ型.....	294
H.2.7 引数の型とデータ型の関係.....	295
H.2.8 復帰値の型.....	296
H.2.9 実行情報の受け渡し.....	297
H.2.10 アプリケーションの呼び出し方法.....	297
H.2.11 アプリケーションとプラグイン定義の例.....	298
H.2.12 プラグイン定義とデータ変換定義の演算式の関係.....	303
付録I プラグイン機能を利用した帳票出力.....	305
I.1 データボックスにデータを配付する.....	305
I.2 帳票データを出力するプラグインの設計.....	305
I.2.1 ifilprints(シングルフォーム出力コマンド).....	305
I.2.2 ifilprintm(マルチフォーム出力コマンド).....	307
付録J サービスバス連携を行う場合のService Integratorの設定.....	317
J.1 サービスバス連携入力を行う場合.....	317
J.2 サービスバス連携出力を行う場合.....	319
付録K プラグイン機能を利用したExcel連携.....	321
K.1 動作環境.....	322
K.1.1 Excelのバージョン.....	322
K.2 利用可能なデータ.....	324
K.2.1 利用可能なデータにおける注意事項.....	325
K.3 Excel連携用プラグインの設計.....	326
K.3.1 ExcelデータをCSVデータに変換するプラグイン.....	327
K.3.2 CSVデータをExcelデータに変換するプラグイン.....	328
K.4 Excel連携で利用するコマンド.....	329
K.4.1 ifiexcelscsv (Excel変換コマンド).....	329
K.4.2 メッセージ.....	332
K.5 定義例.....	334
K.5.1 データ収集したExcelデータをCSVデータへ変換する場合.....	334
K.5.2 データ変換したCSVデータをExcelデータへ変換する場合.....	335
K.6 運用時における注意事項.....	336
K.6.1 プラグイン機能を利用したExcel連携のファンクション実行中の注意事項.....	336

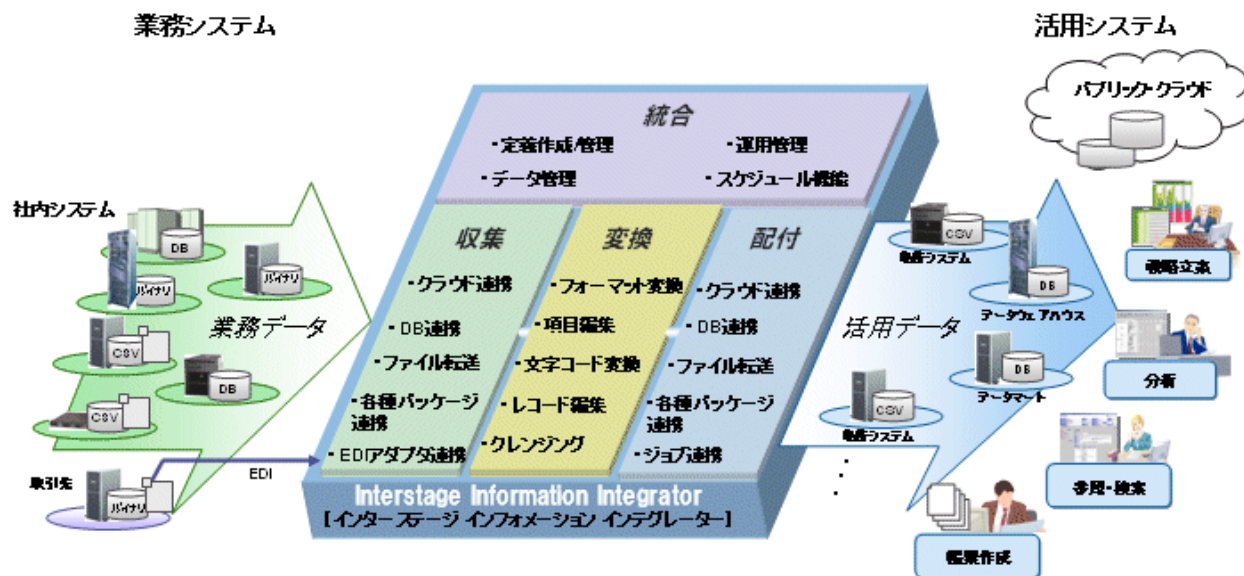
# 第1章 Information Integratorの機能

ここでは、Information Integratorの機能について説明します。

## 1.1 機能概要

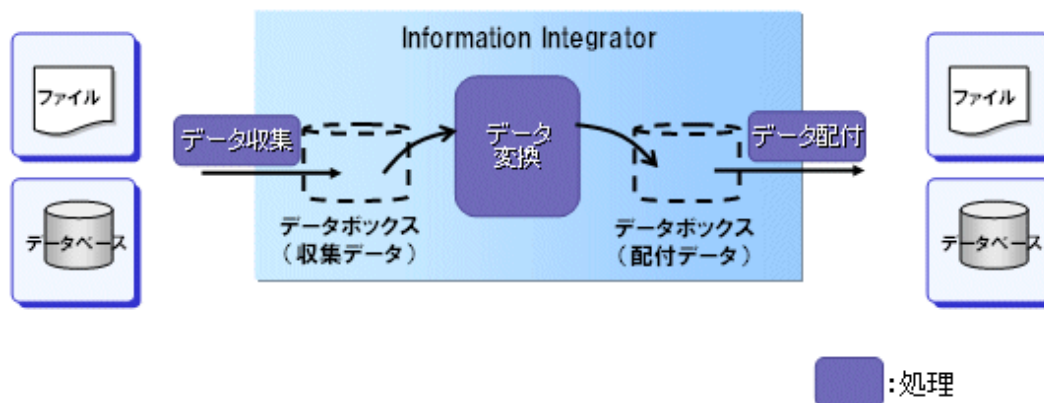
Information Integratorは、多様なデータを収集・統合して、必要な時に必要な形式で配付するデータ統合製品です。企業内のプラットフォーム・運用時間・データ管理方法などが異なる複数の業務システムからデータを集め、データを利用目的に応じた形式に統合して、利用する業務システムに格納することができます。

図1.1 機能概要



Information Integratorでは、データ収集、データ変換、データ配信の一連の処理を以下のような流れで実行します。

図1.2 処理の流れ



データ収集により集めたデータは、Information Integratorのデータ管理機能により、一時的にデータボックスに格納されます。データボックスに格納されたデータに対して、必要な各種変換処理を行い、配付するデータがデータボックス内に作成されます。データボックスに作成された変換処理後のデータを抽出して、データを配付します。

## 1.2 データ収集

データ収集について説明します。



## 1.2.1 Information Integratorで収集可能なデータ形式

---

Information Integratorで収集できるデータ形式を以下に示します。

- ファイル形式
    - CSVファイル
    - バイナリファイル
    - XMLファイル
  - データベース形式
    - Symfoware Server
    - Oracle
    - SQL Server
    - DB2
    - PostgreSQL
    - SQL Azure
    - その他のデータベース(ODBC接続)
  - オブジェクト形式
    - Salesforceオブジェクト

セールスフォース・ドットコム(以降、Salesforceと呼びます)から抽出するオブジェクトです。
  - Information Storage形式
- Information Storageから抽出するデータです。
- BLOB形式
- Azure Storageから収集するデータです。

### ポイント

---

#### 圧縮されたデータの収集

BLOBデータのデータ収集では、zip形式に圧縮されたデータを収集し、収集後にInformation Integratorが提供するプラグインにより解凍して利用することができます。

プラグインの詳細は、“[6.10.3 データを圧縮または解凍する場合](#)”を参照してください。

---

- サービスバス連携形式
    - Service Integrator形式

Service Integratorのサービスバス連携から渡されるデータ形式です。
  - Excelデータ形式
- “[付録K プラグイン機能を利用したExcel連携](#)”を参照してください。

それぞれ、収集可能なデータの形式や、データ型などの条件があります。条件の詳細については、“[B.1 収集できるデータ形式](#)”を参照してください。

## 1.2.2 データ収集方法

---

ファイルの収集では、他システムからファイルを収集するには、「ファイル受信」を、自システムにあるファイルを収集するには、「ファイル入力」を利用します。

データベースから、データを取得するには、「DB抽出」を利用します。



パブリック・クラウドにあるデータの収集では、パブリック・クラウドのオブジェクトを抽出する「オブジェクト抽出」を利用します。  
Information Storageから、データを取得するには、「IIS抽出」を利用します。  
Azure Storageから、データを取得するには、「BLOB収集」を利用します。  
Service Integratorのサービスバス連携のデータを収集するには、「サービスバス連携入力」を利用します。  
Excel形式のデータを収集するには、「Excel連携入力」を利用します。  
それぞれのデータ収集方法について説明します。

### 1.2.3 ファイル受信

ファイル受信について説明します。

ファイル受信は、他システムにあるファイルを収集する場合に、利用する機能です。

ファイル受信には、ファイル受信(自局主導)、ファイル受信(他局主導)の2種類の収集方法があります。

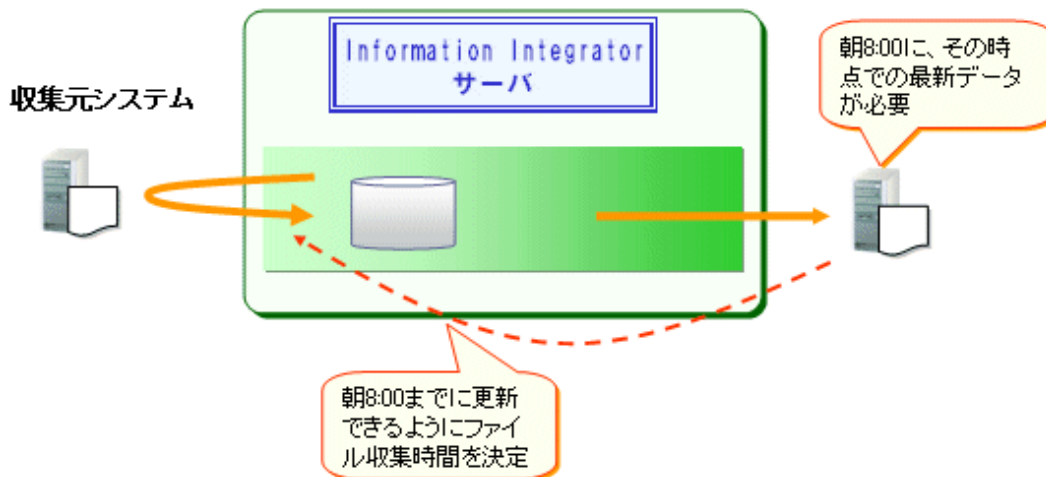
ファイル受信の対象環境については、“4.2 相手側システムの接続について”を参照してください。

#### ファイル受信(自局主導)

Information Integratorから、業務システム(収集元システム)にあるファイルを収集します。

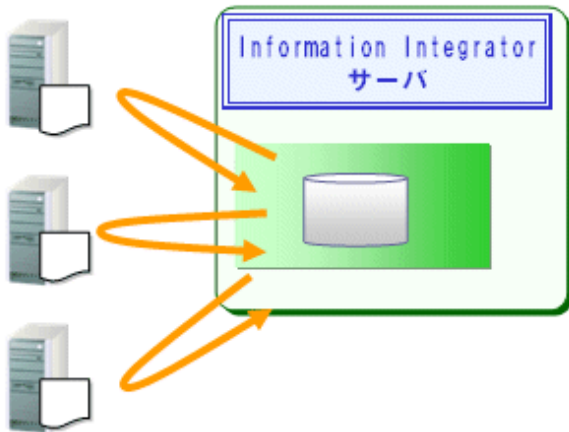
情報活用システム(配付先システム)のスケジュールに合わせて、必要な情報を持ってくる場合に、利用します。例えば、情報活用システム側が、毎朝時点での最新情報を保持しておくことが必要な場合、ファイル受信から、配付先システムに到達する時間を計算し、ファイル受信するスケジュールを指定します。

図1.3 ファイル受信(自局主導)



ファイル受信(自局主導)は、複数の収集元システムにあるファイルを収集することができます。

図1.4 複数の収集元システムからファイルを収集する場合  
収集元システム



### ファイル受信(他局主導)

収集元システムから送られてくるファイルを、Information Integratorで監視し、送られてきたことを確認してから、後続処理を実行します。

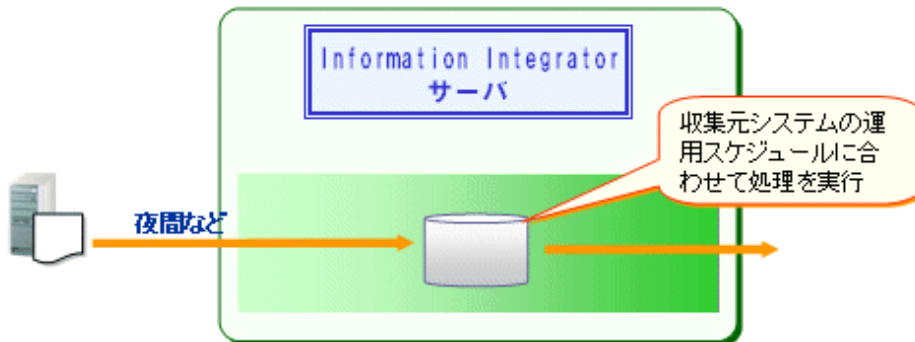
ファイル受信(他局主導)は、収集元システムの運用スケジュールに合わせて、処理を実行できることが特長です。例えば、ファイル受信(他局主導)は、複数の収集元システムから送られてくるデータが揃うのを待つ、または1つの収集元システムから複数のファイルの受信を待つ場合などに利用します。

ファイル受信(他局主導)を利用することで、以下のような運用ができます。

- ・ 収集元システムの運用スケジュールに合わせて処理を実行

収集元システムで必要なデータの準備ができてから、Information Integratorにファイルを送付します。収集元システムで生成される最新のデータを、随時処理できます。

図1.5 収集元システムの運用スケジュールに合わせて処理を実行するパターン

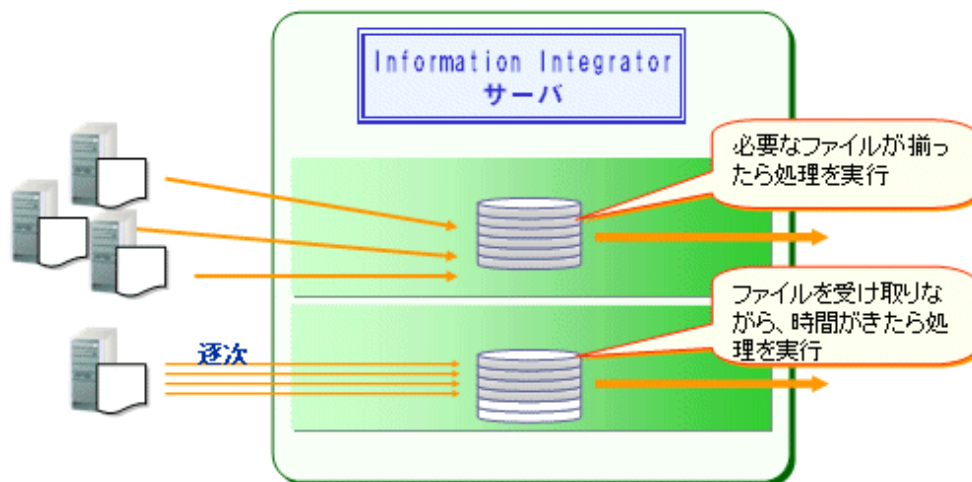


- ・ ファイルの受け取り状況に合わせて後続処理を実行

必要なデータは、収集元システムで準備ができ次第送付し、Information Integratorは、自身の運用スケジュールに合わせて、処理を実行します。最新のデータを受信しつつ、配付先システムの運用に合わせて、データを送付することができます。

また、複数の拠点からのデータを収集する場合や、結合して利用するファイル同士を収集する場合にも利用できます。

図1.6 ファイルの受け取り状況に合わせて後続処理を実行するパターン



### 1.2.3.1 ファイル受信のオプション機能

#### 自動圧縮・自動解凍

ファイル送受信の転送時間は、転送データのサイズに比例します。大規模データの転送や、回線速度の遅いネットワークを利用する転送では、転送時間の増加が顕著になります。

このような場合、ファイルを圧縮、転送後に解凍することで、ネットワーク上に流すデータ量を削減し、転送時間を短縮することができます。

Information Integratorでは、ファイル送受信と連動して自動的に圧縮・解凍します。

圧縮・解凍は、データの転送時間を短縮することが目的です。データ圧縮、解凍の処理速度より、転送速度の方が遅い回線を利用している場合に指定してください。また、転送するデータがすでに圧縮されている場合は、データ圧縮を行わないで転送してください。本機能は、通信プロトコルが、FTP+のときに利用可能です。

#### ポイント

##### 通信プロトコルがFTP+以外の場合 かつ、データ形式がzip形式の場合

zip形式に圧縮されたデータは、収集後にInformation Integratorが提供するプラグインを利用して解凍して、データを扱うことができます。

プラグインの詳細は、“6.10.3 データを圧縮または解凍する場合”を参照してください。

#### 多重度制御

ファイル送受信、相手側ジョブ起動、ジョブ結果通知を行うことによるシステムおよび通信パス単位での回線の同時使用数のことを転送多重度と呼びます。Information Integratorは、この転送多重度の上限値を設定できます。

多くの相手システムとファイル送受信を同時多重実行する場合、Information Integratorサーバの負荷増大に伴いプロトコル応答電文の通知が遅延します。これにより、相手側システムでタイムアウトエラーやネットワークエラーを検出し、ファイル送受信異常が多発する場合があります。多重度制御で同時処理するファイル送受信数を制限することで、受付済みのファイル送受信の応答遅延を防止することができます。

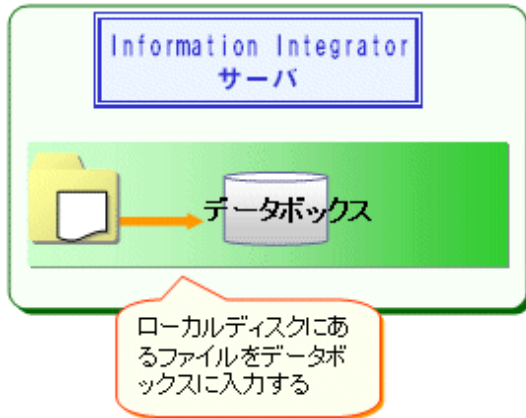
また、HULFTを利用する場合の転送多重度は、HULFT製品の設定に従います。

### 1.2.4 ファイル入力

ローカルディスクにあるファイルを、データボックスに入力します。

データボックスに入力することで、データ変換、データ配付など、後続処理が可能になります。

図1.7 ファイル入力



## 1.2.5 DB抽出

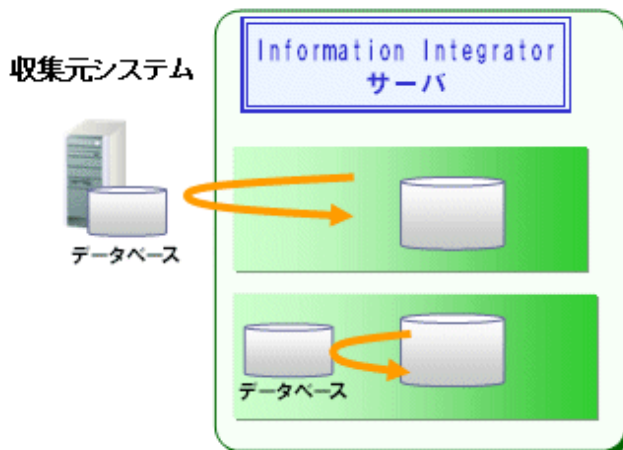
DB抽出は、収集元システムのデータベースに格納されているデータを、Information Integratorサーバ側から抽出する機能です。

また、Information Integratorサーバを導入しているサーバ内のデータベースからも抽出できます。

収集元システムのデータベースに格納されているデータから以下のように抽出します。

- 全件抽出  
データベースのテーブルをまるごと抽出します。
- 条件抽出  
選択したレコードを抽出します。
- 条件ファイル  
Information Integratorサーバ側から抽出条件を記述したファイルを利用して選択したレコードを抽出します。

図1.8 DB抽出



### 順序性制御

データベースから抽出されたデータのトランザクション順で配付したい場合など、収集データの順序性を格納先(配付先システム)まで保証したい場合に、利用します。

- 処理プロセス単位  
処理プロセス単位での順序性を制御します。  
同じ処理プロセスがすでに実行されている場合には、先行する処理プロセスが完了するまで開始されません。

- ・ 待ち合わせ状態の監視

順序性制御により待ち合わせになった処理は、処理可能な状態になるまで待ち合わせますが、待ち合わせ中かどうかは状態監視で確認できます。

状態監視は利用者が常に意識する必要がありますが、一定間隔以上の待ち合わせが発生した場合には以下の警告メッセージを出力しますので、メッセージ監視の可能な製品で監視をすることが可能です。

順序性待ち合わせのプロセスが存在します - プロセスID(~~) サービス番号(~~~)

## 1.2.6 オブジェクト抽出

本機能は、オブジェクトを収集する場合に、利用する機能です。

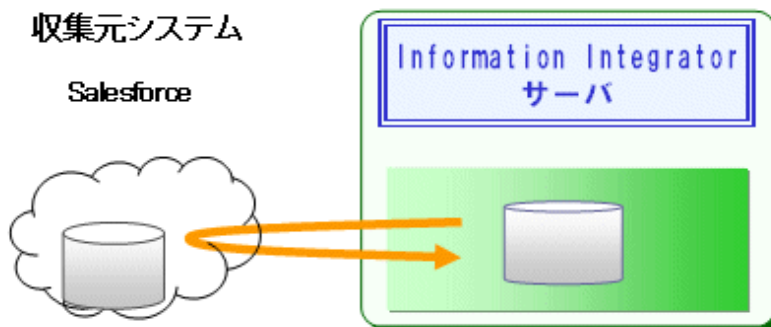
Salesforceオブジェクトの抽出では、収集元システム(Salesforce)に格納されているデータをInformation Integratorサーバ側から抽出します。

Salesforceとの接続形態については、“[4.2.2 オブジェクト抽出・格納時の通信プロトコル](#)”を参照してください。

データ抽出方法は、以下のとおりです。

- ・ 全件抽出  
Information Integratorサーバ側からすべてを抽出します。
- ・ 条件抽出  
Information Integratorサーバ側から選択したレコードを抽出します。
- ・ 条件ファイル  
Information Integratorサーバ側から抽出条件を記述したファイルを利用して選択したレコードを抽出します。

図1.9 データ抽出(自局主導)



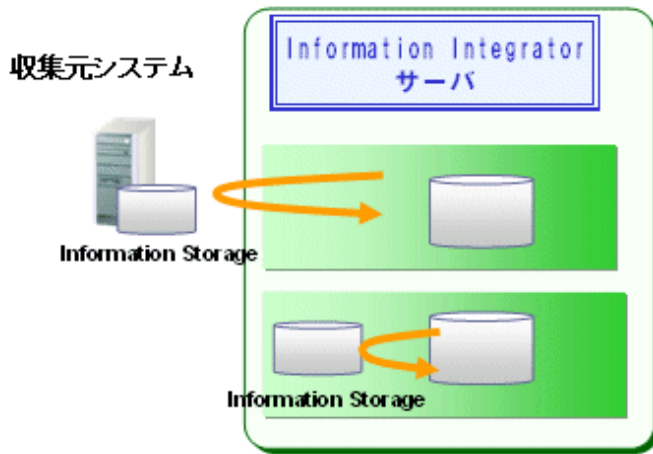
## 1.2.7 IIS抽出

本機能は、Information Storageのデータを抽出する場合に、利用する機能です。

IIS抽出では、収集元システムのInformation Storageに格納されているデータから以下のように抽出します。

- ・ 全件抽出  
Information Integratorサーバ側からすべてを抽出します。
- ・ 条件抽出  
Information Integratorサーバ側から選択したレコードを抽出します。
- ・ 条件ファイル(抽出プロパティファイル)  
Information Integratorサーバ側から抽出条件を記述したファイルを利用して選択したレコードを抽出します。

図1.10 IIS抽出

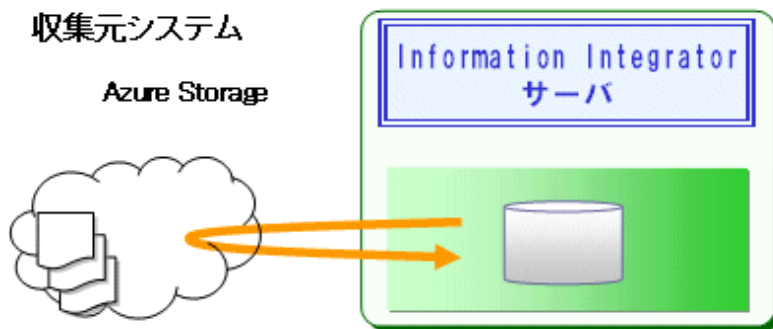


## 1.2.8 BLOB収集

本機能は、Azure Storageのデータを収集する場合に、利用する機能です。

BLOB収集では、収集元システムのAzure Storageに格納されているデータを、Information Integratorサーバ側から収集します。BLOB単位で収集します。

図1.11 BLOB収集

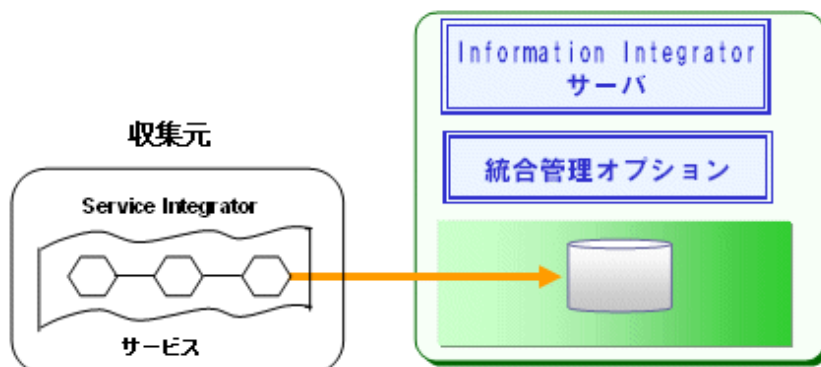


## 1.2.9 サービスバス連携入力

本機能は、サービスバス連携を入力として連携する場合に、利用する機能です。

Service Integratorのサービスバス連携を入力とするには、Service Integratorのアウトバウンドで出力したデータをデータボックスに入力することで、データ変換、データ配付など、後続処理が可能になります。

図1.12 サービスバス連携入力



## ポイント

Service Integratorの定義画面および運用管理画面において、連携するInformation Integratorの定義および運用監視が可能です。

Service Integratorにおける定義方法および運用監視方法については、Service Integratorマニュアルの以下について参照してください。

- ・ 定義

“ISI Studio ヘルプ”において、アウトバウンドのサービスエンドポイントの接続方式に「III連携」を選択した場合のIII連携の定義方法

- ・ 運用監視

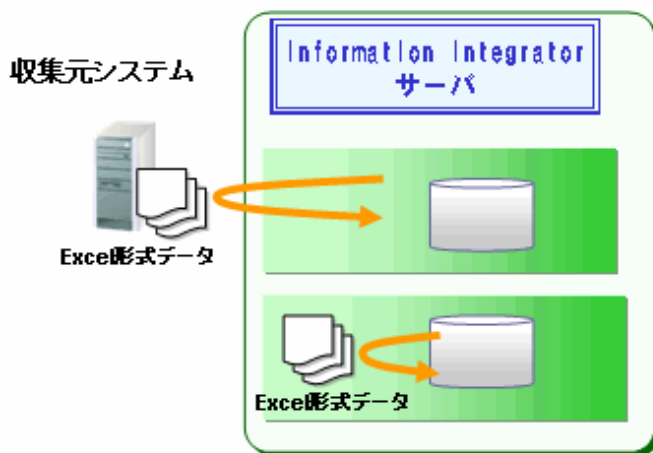
“運用ガイド”および“ISI運用管理コンソールのヘルプ”において、アウトバウンドのサービスエンドポイントのサービスエンドポイント種別が「III連携」の場合に表示されるサービス履歴画面

## 1.2.10 Excel連携入力

本機能は、Excelデータを入力として連携する場合に、利用する機能です。

Excelデータとして扱えるシートの種類や連携方法の詳細は“付録K プラグイン機能を利用したExcel連携”を参照してください。

図1.13 Excel連携入力



## ポイント

Excel連携入力を実行した後に

Information Integrator内ではExcel形式でデータを扱うことはできませんので、Information Integrator内で使用可能なCSVデータに変換する必要があります。

Excelデータを扱うデータ収集の実行後には必ずExcelデータをCSVデータに変換するプラグインを実行してください。

## 1.3 データ配付

データの配付について説明します。

### 1.3.1 Information Integratorで配付可能なデータ形式

Information Integratorで配付できるデータソースを以下に示します。

- ファイル形式
  - CSVファイル
  - バイナリファイル
  - XMLファイル
- データベース形式
  - Symfoware Server
  - Oracle
  - SQL Server
  - DB2
  - PostgreSQL
  - SQL Azure
  - その他のデータベース(ODBC接続)
- オブジェクト形式
  - Salesforceオブジェクト  
Salesforceへ格納するオブジェクトです。
- 帳票データ
  - シングルフォーム
  - マルチフォーム
 Interstage List Creatorと連携して出力する帳票データです。
- Information Storage形式  
Information Storageへ格納するデータです。
- BLOB形式  
Azure Storageへ配付するデータです。

## ポイント

### .....

#### 圧縮されたデータの収集

BLOB形式のデータを送信する前にInformation Integratorが提供するプラグインを利用してzip形式に圧縮してから、データを配付することができます。

プラグインの詳細は、“[6.10.3 データを圧縮または解凍する場合](#)”を参照してください。

.....

- サービスバス連携形式
  - Service Integrator形式  
Service Integratorのサービスバス連携へ渡すデータ形式です。
- Excelデータ形式  
“[付録K プラグイン機能を利用したExcel連携](#)”を参照してください。

それぞれ、配付可能なデータの形式や、データ型などの条件があります。条件の詳細については、“[B.2 配付できるデータ形式](#)”を参照してください。



## 1.3.2 データの配付方法

ファイルの配付では、他システムへファイルを配付するには、「ファイル送信」を、自システムのローカルディスクにファイルを置く場合は、「ファイル出力」を利用します。

データベースへデータを格納するには、「DB格納」を利用します。

パブリック・クラウドへデータを格納するには、「オブジェクト格納」を利用します。

Interstage List Creatorと連携して帳票を出力するには、「帳票出力」を利用します。

Information Storageへデータを格納するには、「IIS格納」を利用します。

Azure Storageへデータを格納するには、「BLOB配付」を利用します。

Service Integratorのサービスバス連携にデータを渡すには、「サービスバス連携出力」を利用します。

Excel形式のデータを出力するには、「Excel連携出力」を利用します。

それぞれのデータ格納方法について説明します。

## 1.3.3 ファイル送信

ファイル送信について説明します。

ファイルを送信する場合のファイルの送信方法は、ファイル送信(自局主導)、ファイル送信(他局主導)の2種類があります。

ファイル送信の対象環境については、「[4.2 相手側システムの接続について](#)」を参照してください。

### ファイル送信(自局主導)

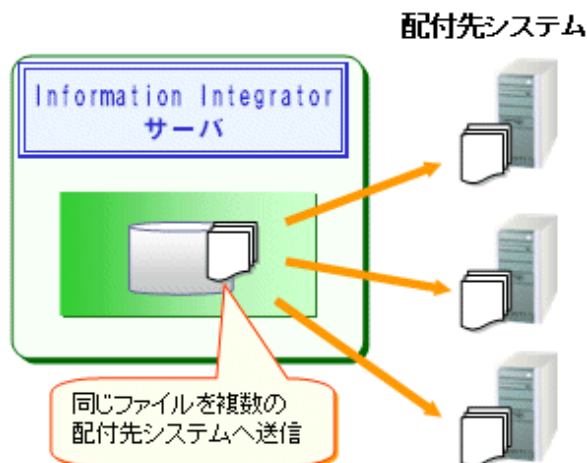
処理が完了したファイルを、配付先システムに送信します。

ファイル送信(自局主導)には、以下の機能があります。

- ・ 同報配信

同一のデータを複数の配付先システムへ送信します。マスタデータの送信業務などに利用します。

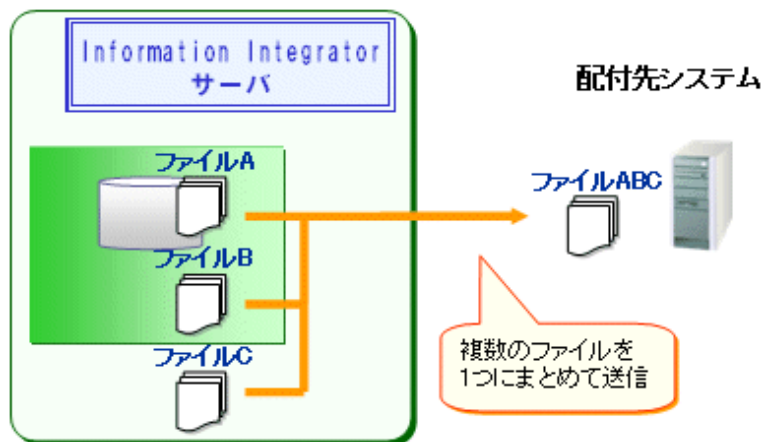
図1.14 同報配信



- 一括配付

処理が完了した複数のファイルを1つのファイルに集約して、配付先システムに送信します。伝票処理で単票データを集約して、送信する場合などに利用します。

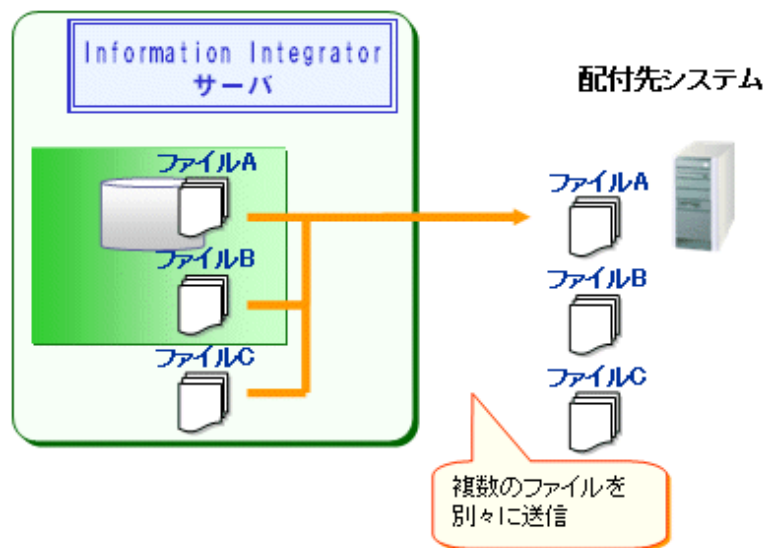
図1.15 一括配付



- 個別配付

処理が完了した複数のファイルを、ファイル単位に、配付先システムに送信します。

図1.16 個別配付

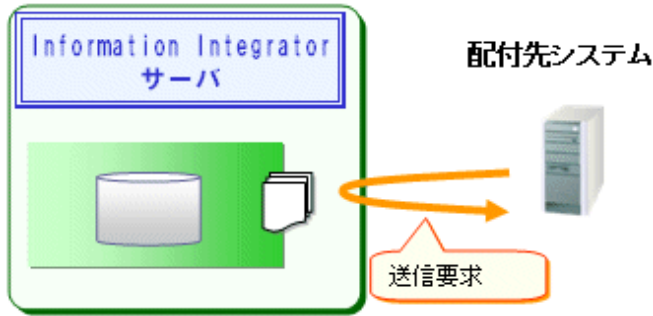


### ファイル送信(他局主導)

配付先システムからの送信要求により、Information Integratorにあるファイルを送信します。

ファイル送信(他局主導)は、配付先システムからの要求を受けた時点で、処理が完了している必要があります。

図1.17 ファイル送信(他局主導)



### 1.3.3.1 ファイル送信のオプション機能

#### 自動圧縮・自動解凍

ファイル送受信の転送時間は、転送データのサイズに比例します。大規模データの転送や、回線速度の遅いネットワークを利用する転送では、転送時間の増加が顕著になります。

このような場合、ファイルを圧縮、転送後に解凍することで、ネットワーク上に流すデータ量を削減し、転送時間を短縮することができます。

Information Integratorでは、ファイル送受信と連動して自動的に圧縮・解凍します。

圧縮・解凍は、データの転送時間を短縮することが目的です。データ圧縮、解凍の処理速度より、転送速度の方が遅い回線を利用している場合に指定してください。また、転送するデータがすでに圧縮されている場合は、データ圧縮を行わないで転送してください。本機能は、通信プロトコルが、FTP+のときに利用可能です。

#### ポイント

##### 通信プロトコルがFTP+以外の場合 かつ、データ形式がzip形式の場合

送信前にInformation Integratorが提供するプラグインを利用してzip形式に圧縮してから、データを配付することができます。

プラグインの詳細は、“[6.10.3 データを圧縮または解凍する場合](#)”を参照してください。

#### 多重度制御

ファイル送受信、ジョブ起動、ジョブ結果通知を行うことによるシステムおよび通信パス単位での回線の同時使用数のことを転送多重度と呼びます。Information Integratorは、この転送多重度の上限値を設定できます。

多くの相手システムとファイル転送を同時多重実行する場合、Information Integratorサーバの負荷増大に伴いプロトコル応答電文の通知が遅延します。これにより、相手側システムでタイムアウトエラーやネットワークエラーを検出し、ファイル送受信異常が多発する場合があります。多重度制御で同時処理するファイル送受信数を制限することで、受付済みのファイル送受信の応答遅延を防止することができます。

また、HULFTを利用する場合の転送多重度は、HULFT製品の設定に従います。

#### 転送速度制御

大容量ファイル転送で、回線速度が遅い場合に、データ転送速度を抑制することで通信路の負荷を軽減します。この場合にデータ転送速度を抑制しなければ、大容量ファイル転送で回線が占有されるため、その他の業務に影響を与えることが考えられます。

#### 注意

ファイル送信時の転送速度の上限を設定する機能であり、必ずしも指定値どおりの転送速度を保証する機能ではありません。システム負荷やネットワーク負荷などの要因により、指定値よりも転送速度が遅くなる場合があります。

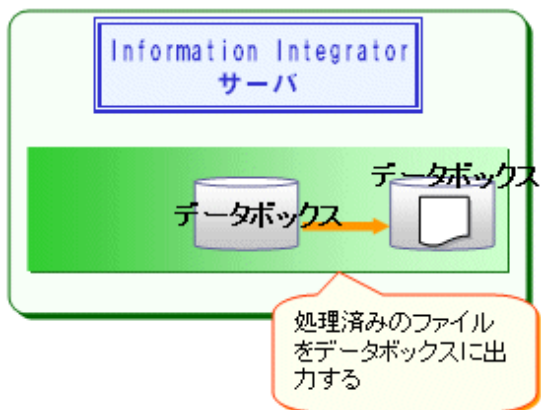
## ファイル置き換え／ファイル追加書き

ファイル送信時に、配付先システムのファイルを置き換えるか、追加書きするかの指定が可能です。

### 1.3.4 ファイル出力

処理(変換など)が完了したファイルを、データボックスに出力します。

図1.18 ファイル出力



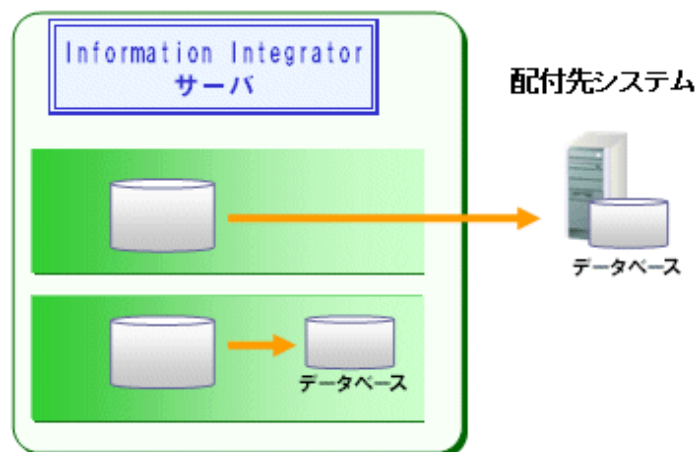
### 1.3.5 DB格納

配付先システムのデータベースに、データを直接格納します。また、複数のテーブルへ、一括で格納することもできます。

DB格納方法には、以下の4種類があります。

- データ置換  
DBのテーブル上のデータをすべて、収集や変換処理の結果データで置換します。
- データ追加  
DBのテーブル上のデータに、収集や変換処理の結果データを追加します。
- データ更新  
DBのテーブル上の該当レコードのすべての項目を、収集や変換処理の結果データで更新します。
- データ追加/更新  
DBのテーブル上の該当レコードのすべての項目を、収集や変換処理の結果データで更新します。該当レコードが存在しない場合にはレコードを追加します。

図1.19 DB格納



## 1.3.6 オブジェクト格納

---

本機能は、オブジェクトを格納する場合に、利用する機能です。

Salesforceオブジェクトの格納では、Information Integratorサーバ側から配付先システム(Salesforce)へ、データを格納します。

Salesforceとの接続形態については、“[4.2.2 オブジェクト抽出・格納時の通信プロトコル](#)”を参照してください。

データ格納方法は、以下のとおりです。

- データ追加

Salesforceのオブジェクトに、収集や変換処理を行った結果のオブジェクトを追加します。

- データ更新

Salesforceにあるオブジェクトの該当するすべての項目を、収集や変換処理を行った結果データで更新します。

- データ追加/更新

Salesforceにあるオブジェクトの該当するすべての項目を、収集や変換処理を行った結果データで更新します。該当項目が存在しない場合はオブジェクトを追加します。

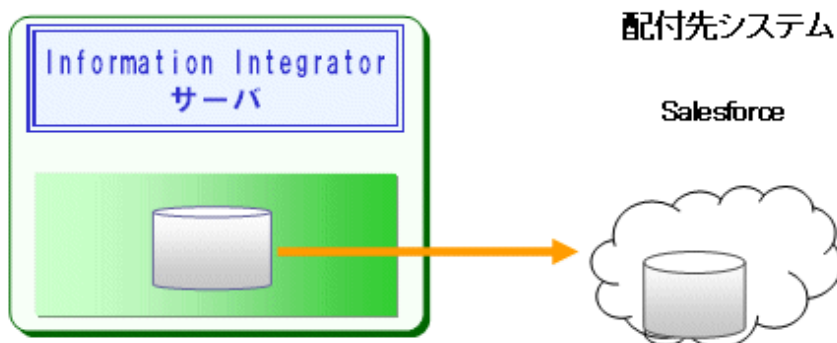
### ポイント

#### データを追加する場合

データを追加する場合、データ追加／更新を利用したオブジェクト格納を推奨します。

データ追加による格納時に何らかの異常が発生した場合、Salesforceのデータのリストアが必要(あらかじめバックアップを実施)となります。

図1.20 Salesforceのオブジェクト格納



### 1.3.6.1 オブジェクト格納のオプション機能

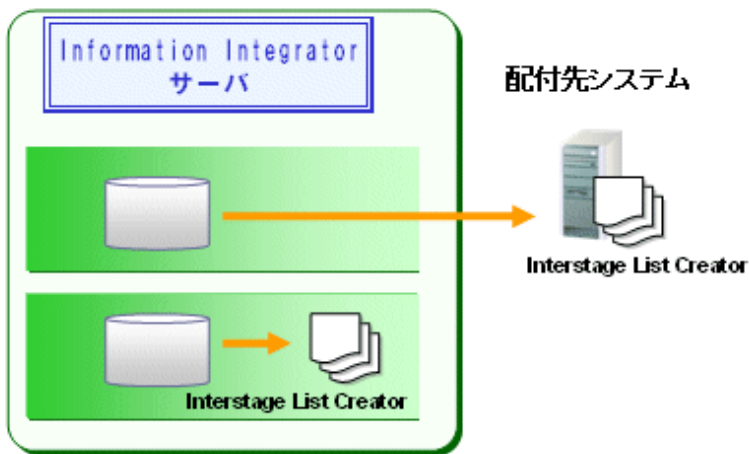
大容量データの転送では、データを分割して転送することができます。詳細は、“[1.3.11 クラウド連携転送データ分割機能](#)”を参照してください。

## 1.3.7 帳票出力

---

Interstage List Creatorと連携して、帳票データを出力する機能です。

図1.21 帳票出力

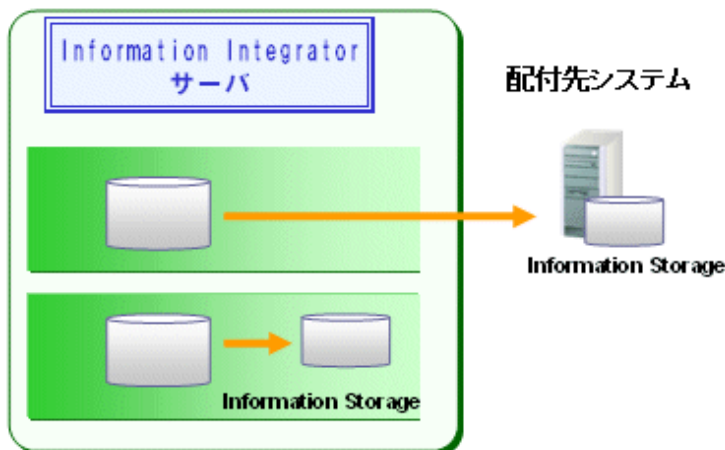


### 1.3.8 IIS格納

Information Storageへデータを格納する場合に利用する機能です。

Information Storageの管理対象ファイル(蓄積データファイル)の形式で格納されます。

図1.22 IIS格納



### 1.3.9 BLOB配付

本機能は、BLOBデータを配付する場合に、利用する機能です。

BLOBデータの配付では、Information Integratorサーバ側から配付先システム(Azure Storage)へ、データを配付する機能です。

Azure Storageとの接続形態については、“4.2.3 BLOB収集・配付時の通信プロトコル”を参照してください。

Azure Storageに該当のBLOBが存在しない場合は、新たにBLOBを作成してBLOBデータを配付します。すでにBLOBが存在する場合は、BLOBデータを上書きして配付します。

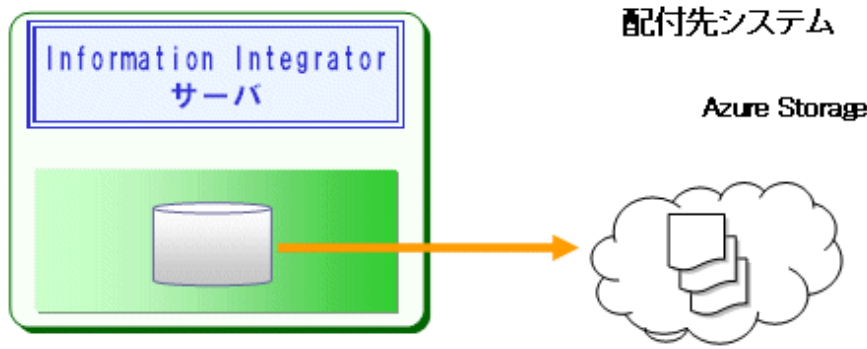
#### ポイント

##### BLOB配付を実行する前に

データの配付時に何らかの異常が発生した場合に備え、あらかじめ何らかの方法によりAzure Storageにあるデータのバックアップを行ってください。

または、データの配付時に何らかの異常が発生した場合に備え、Azure Storageの機能でロールバックを行う運用形態にしてください。

図1.23 BLOB配信



### 1.3.9.1 BLOB配信のオプション機能

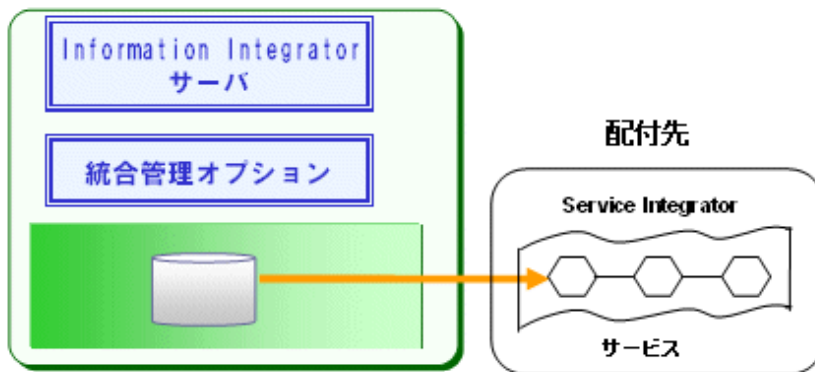
大容量BLOBデータの転送では、データを分割して転送することができます。詳細は、“[1.3.11 クラウド連携転送データ分割機能](#)”を参照してください。

## 1.3.10 サービスバス連携出力

本機能は、サービスバス連携へ出力する場合に、利用する機能です。

Service Integratorのサービスバス連携への出力では、データ配付をService Integratorのインバウンドに連携させることで、Service Integratorへの出力に利用する機能です。

図1.24 サービスバス連携出力



### ポイント

Service Integratorの定義画面および運用管理画面において、連携するInformation Integratorの定義および運用監視が可能です。

Service Integratorにおける定義方法および運用監視方法については、Service Integratorマニュアルの以下について参照してください。

- ・ 定義  
“ISI Studio ヘルプ”において、インバウンドのサービスエンドポイントの接続方式に「III連携」を選択した場合のIII連携の定義方法
- ・ 運用監視  
“運用ガイド”および“ISI運用管理コンソールのヘルプ”において、インバウンドのサービスエンドポイントのサービスエンドポイント種別が「III連携」の場合に表示されるサービス履歴画面

### 1.3.11 クラウド連携転送データ分割機能

---

大容量データの転送では、データを分割して転送することができます。これにより、通信路の負荷を軽減します。

データ転送量を抑止しないと、大容量データ転送では回線が占有され、その他の業務に影響を与えることが考えられます。

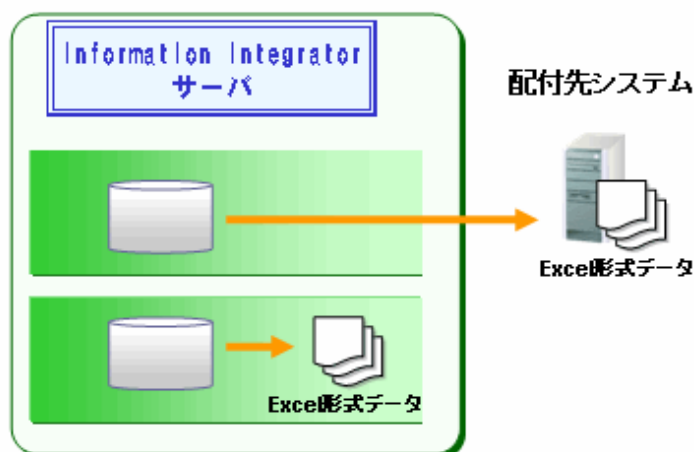
### 1.3.12 Excel連携出力

---

Excelデータを出力する機能です。

Excelデータとして扱えるシートの種類や連携方法の詳細は“[付録K プラグイン機能を利用したExcel連携](#)”を参照してください。

図1.25 Excel連携出力



#### **P** ポイント

Excel連携出力を実行する前に

Information Integrator内ではExcel形式でデータを扱うことはできませんので、CSVデータで処理を行います。Excel連携出力を行うには必ず該当のデータ配付前に任意のCSVデータをExcelデータに変換するプラグインを実行してください。

## 1.4 データ変換・編集

---

Information Integratorが収集したデータを変換・編集することができます。

なお、収集したデータの特定レコードや特定の項目だけを変換・編集の対象にすることもできます。

Information Integratorのデータ変換・編集機能は、以下の4種類があります。

- ・ ファイル形式変換
- ・ 文字コード変換
- ・ 項目編集
- ・ レコード編集



## ポイント

- Information Integratorのデータ変換・編集機能では、変換・編集時に変換前、変換後の各データ(CSVファイルまたはバイナリファイル)に対して、それぞれ以下のデータチェックを行います。異常を検知すると、エラーメッセージを出力します。

- データ形式

以下の入力値チェックを行います。

- 全角文字列属性
- 日付・時間属性
- 数値属性
- バイナリ属性
- ビット属性

- 桁数の過不足に対するデータチェック

- 必須入力チェック

各データチェックのチェック内容については、“付録F 変換・編集時に行うデータチェックについて”を参照してください。

## 注意

- 変換前または変換後のデータがXMLファイルの場合、XMLファイルに対してのデータチェックは行いません。
- Salesforceから以下のデータ型のデータを抽出した場合、収集した当該データ(変換前データ)に対してのデータチェックを行うことはできません。

- int
- double
- currency
- percent

- 既存システムに使用していた変換アプリケーションをプラグイン機能により、Information Integratorに組み込むことができます。プラグイン機能についての詳細は、“1.7 プラグイン”を参照してください。
- 変換後のデータをCSVファイルまたはXMLファイルで扱う場合、文字型のデータだけに囲み文字(ダブルクォーテーションのみ)を付与する、または、すべてのデータに囲み文字を付与しないのいずれかを選択することができます。

## 1.4.1 ファイル形式変換

ファイル形式を別のファイル形式に変換する機能です。収集したデータ全体を対象に変換します。バイナリファイル、CSVファイル、XMLファイルの相互フォーマット変換ができます。

## 注意

### XMLファイルの変換について

Information IntegratorでXMLファイルからXMLファイルへ変換する場合は、一度XMLファイルをCSVファイル形式に変換した後、再度XMLファイルへの変換が必要です。

### データ変換時のエラー処理について

ファイル形式変換、標準コード変換の場合、データ変換時のエラー後の動作を選択することができます。

以下に、それぞれ選択できる範囲を示します。

表1.1 データ変換時のエラー原因とエラー後の動作

属性	エラー原因	動作	エラー後の動作(メッセージ)
文字列演算	文字コード変換	中断	中断(エラー)
		継続	代替文字を入れて処理継続(警告)
	文字あふれ	中断	中断(エラー)
		継続	あふれた文字を削除して処理継続(警告)
内部10進	先頭4bitが0以外	中断	中断(エラー)
		継続	先頭4bitを「0」にして処理継続(警告)
数値演算	桁あふれ	中断	中断(エラー)
	0除算	中断	中断(エラー)

## 1.4.2 文字コード変換

指定された文字コードから、別の文字コードに変換する機能です。

収集したデータ全体を対象に変換します。

Information Integratorでは、文字コード変換の標準機能としてJIS第一水準、JIS第二水準の文字コード変換が可能です。変換可能な組合せは、以下のとおりです。

表1.2 文字コード変換可能な組合せ

入力文字コード		出力文字コード														
		Shift-JIS	EUC			JEF-EBCDIC				Unicode						
			U90	S90	HP	カナ (字形重視)	カナ (領域重視)	ASCII (字形重視)	ASCII (領域重視)	UCS 2	UCS 2(MS )	CFS D (注2)	UTF 8	UTF 8-4	UTF 16	UTF 16(M S)
Shift-JIS	○ (注1)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	
EUC	U90	○	—	—	—	○	○	○	○	○	○	○	○	○	○	
	S90	○	—	○ (注1)	—	○	○	○	○	○	○	○	○	○	○	
	HP	○	—	—	—	○	○	○	○	—	—	—	—	—	—	
JEF-EBCDIC	カナ (字形重視)	○	○	○	○	—	—	—	—	○	○	○	○	○	○	
	カナ (領域重視)	○	○	○	○	—	—	—	—	—	—	—	—	—	—	
	ASCII (字形重視)	○	○	○	○	—	—	—	—	○	○	○	○	○	○	
	ASCII (領域重視)	○	○	○	○	—	—	—	—	—	—	—	—	—	—	
Unicode	UCS2	○	○	○	—	○	—	○	—	○ (注1)	○	○	○	—	—	

入力文字コード		出力文字コード														
		Shift-JIS	EUC			JEF-EBCDIC				Unicode						
			U90	S90	HP	カナ (字形重視)	カナ (領域重視)	ASCII (字形重視)	ASCII (領域重視)	UCS 2	UCS 2(MS)	CFS D (注2)	UTF 8	UTF 8-4	UTF 16	UTF 16(M S)
	UCS2( MS)	○	○	○	—	○	—	○	—	○	○ (注1)	○	○	—	—	—
	CFS D (注2)	○	○	○	—	○	—	○	—	○	○ (注1)	○	○	—	—	—
	UTF8	○	○	○	—	○	—	○	—	○	○	○	○ (注1)	—	—	—
	UTF8-4	○	○	○	—	○	—	○	—	○	○	○	○	○ (注1)	○	○
	UTF16	○	○	○	—	○	—	○	—	○	○	○	○	○	○ (注1)	○
	UTF16 (MS)	○	○	○	—	○	—	○	—	○	○	○	○	○	○	○ (注1)
IBM日 本語 文字 セット	カナ	○	○	○	—	○	○	—	—	—	—	—	—	—	—	—
	英小文 字	○	○	○	—	—	—	—	—	—	—	—	—	—	—	—

○:変換可能です。—:変換不可能な組み合わせ

注1)数値の表現形式を変更する場合に指定できます。

注2)ifcnvdatコマンドでは指定できません。

## ポイント

JEFの旧字体(「塚」の旧字など)やベンダ固有文字(「株」など)は、文字コード変換テーブルを切り替えることで、文字コード変換できます。また、外字の文字コード変換については、文字コード変換テーブルをカスタマイズすることで、文字コード変換できます。

文字コード変換テーブルの切替え、およびカスタマイズ方法については、“[第10章 文字コード変換の準備](#)”を参照してください。

## 注意

Information Integratorが動作するシステムにCharset Managerが導入されている場合で、Charset Managerの文字コード変換表をカスタマイズしている場合は、この文字コード変換表をInformation Integratorの文字コード変換テーブルに取り込んでください。詳細は、“[第10章 文字コード変換の準備](#)”、および“[コマンドリファレンス](#)”を参照してください。

## データ変換時のエラー処理について

ファイル形式変換、文字コード変換の場合、データ変換時のエラー後の動作を選択することができます。

以下に、それぞれ選択できる範囲を示します。

表1.3 データ変換時のエラー原因とエラー後の動作

属性	エラー原因	動作	エラー後の動作(メッセージ)
文字列演算	文字コード変換	中断	中断(エラー)
		継続	代替文字を入れて処理継続(警告)
	文字あふれ	中断	中断(エラー)
		継続	あふれた文字を削除して処理継続(警告)
内部10進	先頭4bitが0以外	中断	中断(エラー)
		継続	先頭4bitを「0」にして処理継続(警告)
数値演算	桁あふれ	中断	中断(エラー)
	0除算	中断	中断(エラー)

### 1.4.3 項目編集

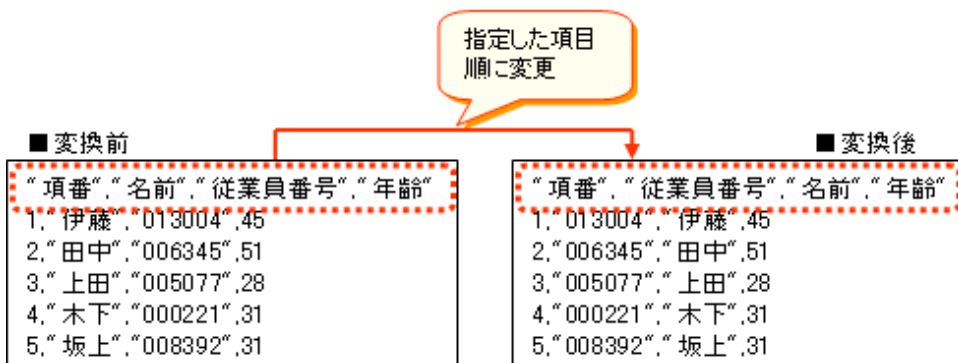
データの項目に対して、編集できます。

#### 1.4.3.1 項目並び替え

項目を指定した順番に並び替えることができます。

【例】

元データの項目「項番、名前、従業員番号、年齢」を、「項番、従業員番号、名前、年齢」の順番に並び替えることができます。

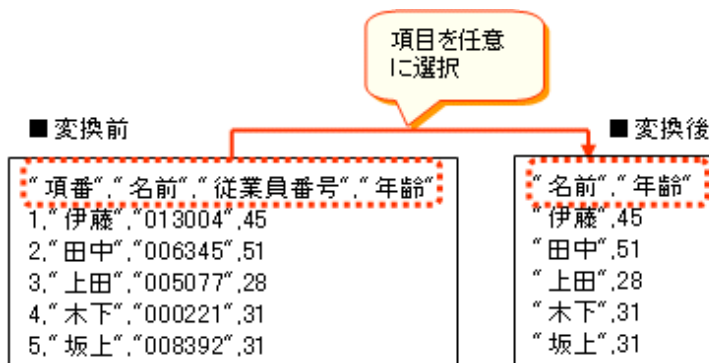


#### 1.4.3.2 項目選択

元データの項目に対して、指定した項目のみを出力することができます。

【例】

元データの項目(項番、名前、従業員番号、年齢)の中から、出力する項目(名前、年齢)を選択することができます。出力データには、その指定した項目が出力されます。



### 1.4.3.3 項目追加

元データにない項目値(定数や演算結果など)を新しい項目に格納することができます

#### 【例】

元データにない項目(利用ポイント)に定数値0を格納する例を示します。

その他にも、文字列の部分抽出などの演算結果を、新しい項目に格納することができます。

■ 変換前

```

顧客 会員番号 年齢
佐藤 013004 45
中田 006345 51
田上 005077 28
斉藤 000221 31
鈴木 008392 31

```

■ 変換後

```

顧客 会員番号 年齢 利用ポイント
佐藤 013004 45 0
中田 006345 51 0
田上 005077 28 0
斉藤 000221 31 0
鈴木 008392 31 0

```

定数値「0」を  
追加

### 1.4.3.4 属性変換

入力側と出力側のデータ型の型変換を行い、出力側のデータ型に従ってデータを変換する機能です。なお、レコードタイプがバイナリの場合に有効な機能です。

なお、収集元や配信先データとInformation Integrator内で扱うデータ型との対応関係については、“[付録B Information Integratorで扱えるデータ形式について](#)”を参照してください。

表1.4 属性変換マトリクス

indata	outdata															
	SQL_CHAR	SQL_VARCHAR	SQL_NCHAR	SQL_VARCHAR	SQL_INTEGER	SQL_SMALLINT	SQL_NUMERIC	SQL_DECIMAL	SQL_REAL	SQL_FLOAT	SQL_DOUBLE	SQL_TIMESTAMP	SQL_TIMESTAMP	SQL_BINARY	SQL_BINARY	
SQL_CHAR	○	○	△	△	○	○	○	○	○	○	○	○	○	-	-	-
SQL_VARCHAR	○	○	△	△	○	○	○	○	○	○	○	○	○	-	-	-
SQL_NCHAR	△	△	○	○	△	△	△	△	-	-	-	-	-	-	-	-
SQL_VARNCHAR	△	△	○	○	△	△	△	△	-	-	-	-	-	-	-	-
SQL_INTEGER	○	○	△	△	○	○	○	○	-	-	-	-	-	-	-	-
SQL_SMALLINT	○	○	△	△	○	○	○	○	-	-	-	-	-	-	-	-
SQL_NUMERIC	○	○	△	△	○	○	○	○	-	-	-	-	-	-	-	-
SQL_DECIMAL	○	○	△	△	○	○	○	○	-	-	-	-	-	-	-	-
SQL_REAL	○	○	-	-	-	-	-	-	○	○	-	-	-	-	-	-

indata	outdata															
	SQL_CH AR	SQL_VA RC HA R	SQL_NC HA R	SQL_VA RN CH AR	SQL_IN TE GE R	SQL_S MA LLI NT	SQL_NU ME RIC	SQL_DE CIMAL	SQL_RE AL	SQL_FL OAT	SQL_DA TE	SQL_TI ME	SQL_TI ME STA MP	SQL_LO NG VA RBI NARY	SQL_BI NARY	SQL_BI T
SQL_FLOAT	○ 79	○ 80	-	-	-	-	-	-	○ 81	○ 82	-	-	-	-	-	-
SQL_DATE	○ 83	○ 84	-	-	-	-	-	-	-	-	○ 85	-	-	-	-	-
SQL_TIME	○ 86	○ 87	-	-	-	-	-	-	-	-	-	○ 88	-	-	-	-
SQL_TIMESTAMP	○ 89	○ 90	-	-	-	-	-	-	-	-	-	-	○ 91	-	-	-
SQL_LONGVARIABLE	-	-	-	-	-	-	-	-	-	-	-	-	-	○ 92	-	-
SQL_BINARY	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○ 93	-
SQL_BIT	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	○ 94

○:変換可能です

△:データ変換定義の「演算式-演算種別」に「全角/半角変換」、「演算式-引数」に「全角変換」または「半角変換」を指定することで、変換可能です

-:変換できません

数値情報は、「属性変換データ例」の「項番」と合致しています。

## ポイント

### 属性変換時の留意事項

- SQL\_LONGVARIABLE型およびSQL\_BINARY型では、バイナリファイルーCSVファイル間での変換は以下のようになります。
  - バイナリファイルからCSVファイルへ変換するとき  
バイナリデータは、BASE64変換によりCSVファイルのデータに変換します。
  - CSVファイルからバイナリファイルへ変換するとき  
バイナリファイルからCSVファイルへBASE64変換により変換したCSVファイルのデータを、バイナリデータに変換します。指定された全体桁に満たないデータは、0をパディングします。
- SQL\_BIT型で、バイナリファイルーCSVファイル間での変換は以下のようになります。
  - バイナリファイルからCSVファイルへ変換するとき
    - バイナリデータは、2進数で表現された文字列(半角文字の0または、1)に変換します。
    - 指定された全体桁が8の倍数にならない場合、CSVファイルの入力データの末尾にあるデータを無視します。

例えば、全体桁の定義を6桁としている場合、バイナリデータが「0x0A (ビット列) 00001010」のときは、CSVファイルへの変換後のデータは、「000010」となります。

ー CSVファイルからバイナリファイルへ変換するとき

- CSVファイルの2進数で表現された文字列(半角文字の0または、1)は、バイナリデータに変換します。
- 指定された全体桁が8の倍数にならない場合、バイナリファイルの出力データの末尾に、0をパディングします。

例えば、全体桁の定義を6桁としている場合、CSVファイルのデータが、「000010」のときは、バイナリファイルへの変換後のデータは、「0x0A (ビット列) 00001010」となります。

属性変換マトリクスで○、または△で変換可能なパターンについて、入力データの変換後のデータ(出力データ)例を以下に示します。「項番」は、“属性変換マトリクス”の数値情報と合致しています。

表1.5 属性変換データ例

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考
1	文字属性	0x31323334	SQL_CHAR(4)	0x31323334	SQL_CHAR(4)	1234の文字
2				0x040031323334	SQL_VARCHAR(4)	1234の文字
3				0x8250825182528253	SQL_NCHAR	1234の文字
4				0x04008250825182528253	SQL_VARNCHAR	1234の文字
5				0xD2040000	SQL_INTEGER	1234の数値
6				0xD204	SQL_SMALLINT	1234の数値
7				0x31323334	SQL_NUMERIC	符号なし
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
8				0x012343	SQL_DECIMAL	1234の数値
9				0x449a4000	SQL_REAL	1234の数値
10				0x4093480000000000	SQL_FLOAT	1234の数値
11	0x323030392F30372F3031	SQL_CHAR(10)	0xD9070701	SQL_DATE	2009/07/01の日付	
12	0x31323A33343A3536	SQL_CHAR(8)	0x0C2238	SQL_TIME	12:34:56の時間	
13	0x323030392F30372F30312031323A33343A3536	SQL_CHAR(19)	0xD90707010C2238	SQL_TIMESTAMP	2009/07/01 12:34:56の日付時間	
14	0x040031323334	SQL_VARCHAR(4)	0x31323334	SQL_CHAR(4)	1234の文字	

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考		
15				0x040031323334	SQL_VARCHAR(4)	1234の文字		
16				0x8250825182528253	SQL_NCHAR	1234の文字		
17				0x04008250825182528253	SQL_VARCHAR	1234の文字		
18				0xD2040000	SQL_INTEGER	1234の数値		
19				0xD204	SQL_SMALLINT	1234の数値		
20				0x31323334	SQL_NUMERIC	符号なし		
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式		
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式		
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式		
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式		
21				0x012343	SQL_DECIMAL	1234の数値		
22				0x449a4000	SQL_REAL	1234の数値		
23				0x4093480000000000	SQL_FLOAT	1234の数値		
24				0x0a00323030392F30372F3031	SQL_VARCHAR(10)	0xD9070701	SQL_DATE	2009/07/01の日付
25				0x080031323A33343A3536	SQL_VARCHAR(8)	0x0C2238	SQL_TIME	12:34:56の時間
26				0x1300323030392F30372F30312031323A33343A3536	SQL_VARCHAR(19)	0xD90707010C2238	SQL_TIMESTAMP	2009/07/01 12:34:56の日付時間
27				0x8250825182528253	SQL_NCHAR(4)	0x31323334	SQL_CHAR(4)	1234の文字
28						0x040031323334	SQL_VARCHAR(4)	1234の文字
29						0x8250825182528253	SQL_NCHAR	1234の文字
30						0x04008250825182528253	SQL_VARCHAR	1234の文字
31						0xD2040000	SQL_INTEGER	1234の数値
32						0xD204	SQL_SMALLINT	1234の数値
33						0x31323334	SQL_NUMERIC	符号なし



項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考	
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式	
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式	
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式	
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式	
34				0x012343	SQL_DECIMAL	1234の数値	
35	0x04008250825182528253	SQL_VARCHAR(4)		0x31323334	SQL_CHAR(4)	1234の文字	
36			0x040031323334	SQL_VARCHAR(4)	1234の文字		
37			0x8250825182528253	SQL_NCHAR	1234の文字		
38			0x04008250825182528253	SQL_VARCHAR	1234の文字		
39			0xD2040000	SQL_INTEGER	1234の数値		
40			0xD204	SQL_SMALLINT	1234の数値		
41			0x31323334	SQL_NUMERIC	符号なし		
			0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式		
			0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式		
			0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式		
					0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
42					0x012343	SQL_DECIMAL	1234の数値
43	数値属性	0xD2040000	SQL_INTEGER	0x31323334	SQL_CHAR(4)	1234の文字	
44				0x040031323334	SQL_VARCHAR(4)	1234の文字	
45				0x8250825182528253	SQL_NCHAR	1234の文字	
46				0x04008250825182528253	SQL_VARCHAR	1234の文字	
47				0xD2040000	SQL_INTEGER	1234の数値	
48				0xD204	SQL_SMALLINT	1234の数値	

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考			
49				0x31323334	SQL_NUMERIC	符号なし			
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式			
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式			
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式			
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式			
50				0x012343	SQL_DECIMAL	1234の数値			
51	0xD204		SQL_SMALLINT	0x31323334	SQL_CHAR(4)	1234の文字			
52				0x040031323334	SQL_VARCHAR(4)	1234の文字			
53				0x8250825182528253	SQL_NCHAR	1234の文字			
54				0x04008250825182528253	SQL_VARCHAR	1234の文字			
55				0xD2040000	SQL_INTEGER	1234の数値			
56				0xD204	SQL_SMALLINT	1234の数値			
57							0x31323334	SQL_NUMERIC	符号なし
							0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式
							0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式
							0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式
							0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
58							0x012343	SQL_DECIMAL	1234の数値
59	0x31323334		SQL_NUMERIC 符号なし	0x31323334	SQL_CHAR(4)	1234の文字			
60				0x040031323334	SQL_VARCHAR(4)	1234の文字			
61				0x8250825182528253	SQL_NCHAR	1234の文字			
62				0x04008250825182528253	SQL_VARCHAR	1234の文字			
63				0xD2040000	SQL_INTEGER	1234の数値			

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考	
64				0xD204	SQL_SMALLINT	1234の数値	
65				0x31323334	SQL_NUMERIC	符号なし	
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式	
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式	
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式	
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式	
0x012343		SQL_DECIMAL	1234の数値				
66		0x31323344		SQL_NUMERIC	0x31323334	SQL_CHAR(4)	1234の文字
					0x040031323334	SQL_VARCHAR(4)	1234の文字
					0x8250825182528253	SQL_NCHAR	1234の文字
					0x04008250825182528253	SQL_VARCHAR	1234の文字
					0xD2040000	SQL_INTEGER	1234の数値
					0xD204	SQL_SMALLINT	1234の数値
	0x31323334	SQL_NUMERIC			符号なし		
	0x31323344	SQL_NUMERIC			符号あり、末尾、ビット形式		
	0x41323334	SQL_NUMERIC			符号あり、先頭、ビット形式		
	0x2B31323334	SQL_NUMERIC			符号あり、先頭、文字形式		
	0x313233342B	SQL_NUMERIC			符号あり、末尾。文字形式		
	0x012343	SQL_DECIMAL			1234の数値		
0x41323334	0x31323334	SQL_CHAR(4)	1234の文字				
	0x040031323334	SQL_VARCHAR(4)	1234の文字				
	0x8250825182528253	SQL_NCHAR	1234の文字				
	0x04008250825182528253	SQL_VARCHAR	1234の文字				

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考	
				0xD2040000	SQL_INTEGER	1234の数値	
				0xD204	SQL_SMALLINT	1234の数値	
				0x31323334	SQL_NUMERIC	符号なし	
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式	
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式	
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式	
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式	
				0x012343	SQL_DECIMAL	1234の数値	
		0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式	0x31323334	SQL_CHAR(4)	1234の文字
					0x040031323334	SQL_VARCHAR(4)	1234の文字
					0x8250825182528253	SQL_NCHAR	1234の文字
					0x04008250825182528253	SQL_VARCHAR	1234の文字
					0xD2040000	SQL_INTEGER	1234の数値
					0xD204	SQL_SMALLINT	1234の数値
					0x31323334	SQL_NUMERIC	符号なし
					0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式
					0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式
					0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式
					0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
					0x012343	SQL_DECIMAL	1234の数値
		0x313233342B	SQL_NUMERIC	符号あり、末尾、文字形式	0x31323334	SQL_CHAR(4)	1234の文字
0x040031323334	SQL_VARCHAR(4)				1234の文字		
0x8250825182528253	SQL_NCHAR				1234の文字		

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考
				0x04008250825182528253	SQL_VARNCHAR	1234の文字
				0xD2040000	SQL_INTEGER	1234の数値
				0xD204	SQL_SMALLINT	1234の数値
				0x31323334	SQL_NUMERIC	符号なし
				0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式
				0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式
				0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式
				0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
				0x012343	SQL_DECIMAL	1234の数値
67				0x31323334	SQL_CHAR(4)	1234の文字
68				0x040031323334	SQL_VARCHAR(4)	1234の文字
69				0x8250825182528253	SQL_NCHAR	1234の文字
70				0x04008250825182528253	SQL_VARNCHAR	1234の文字
71				0xD2040000	SQL_INTEGER	1234の数値
72				0xD204	SQL_SMALLINT	1234の数値
		0x012343	SQL_DECIMAL	0x31323334	SQL_NUMERIC	符号なし
		0x012343	SQL_DECIMAL	0x31323344	SQL_NUMERIC	符号あり、末尾、ビット形式
73		0x012343	SQL_DECIMAL	0x41323334	SQL_NUMERIC	符号あり、先頭、ビット形式
		0x012343	SQL_DECIMAL	0x2B31323334	SQL_NUMERIC	符号あり、先頭、文字形式
		0x012343	SQL_DECIMAL	0x313233342B	SQL_NUMERIC	符号あり、末尾。文字形式
74		0x012343	SQL_DECIMAL	0x012343	SQL_DECIMAL	1234の数値
75				0x31323334	SQL_CHAR(4)	1234の文字
76		0x449a4000	SQL_REAL	0x040031323334	SQL_VARCHAR(4)	1234の文字
77		0x449a4000	SQL_REAL	0x449a4000	SQL_REAL	1234の数値

項番	分類	入力データ	入力データのデータ型	出力データ	出力データのデータ型	備考
78		0x40934800000000	SQL_FLOAT	0x4093480000000000	SQL_FLOAT	1234の数値
79				0x31323334	SQL_CHAR(4)	1234の文字
80				0x040031323334	SQL_VARCHAR(4)	1234の文字
81				0x449a4000	SQL_REAL	1234の数値
82				0x4093480000000000	SQL_FLOAT	1234の数値
83	日	0xD9070701	SQL_DATE	0x323030392F30372F3031	SQL_CHAR(10)	2009/07/01の日付
84				0x0a00323030392F30372F3031	SQL_VARCHAR(10)	2009/07/01の日付
85				0xD9070701	SQL_DATE	
86	付時	0x0C2238	SQL_TIME	0x31323A33343A3536	SQL_CHAR(8)	12:34:56の時間
87				0x080031323A33343A3536	SQL_VARCHAR(8)	12:34:56の時間
88				0x0C2238	SQL_TIME	
89	間属性	0xD90707010C2238	SQL_TIMESTAMP	0x323030392F30372F30312031323A33343A3536	SQL_CHAR(19)	2009/07/01 12:34:56の日付時間
90				0x1300323030392F30372F30312031323A33343A3536	SQL_VARCHAR(19)	2009/07/01 12:34:56の日付時間
91				0xD90707010C2238	SQL_TIMESTAMP	
92		0x000000002000001234	SQL_LONGVARBINARY(2)	0x000000002000001234	SQL_LONGVARBINARY(2)	BLOB形式
93		0x0102030405	SQL_BINARY(5)	0x0102030405	SQL_BINARY(5)	
94		0x0a	SQL_BIT(8)	0x0a	SQL_BIT(8)	
			SQL_BIT(6)	0x08	SQL_BIT(6)	

### 1.4.3.5 トリム

文字列から1文字以上連続する任意の文字をすべて取り除きます。トリムでは、以下の2種類が利用できます。

- ・ ライトトリム

文字列の末尾にある1文字以上連続する任意の文字をすべて取り除きます。

データベースからデータを抽出した際、char型属性の項目は、文字列の末尾に空白文字が埋め込まれているため、これらを取り除く時に利用できます。

- ・ レフトトリム

文字列の冒頭にある1文字以上連続する任意の文字をすべて取り除きます。

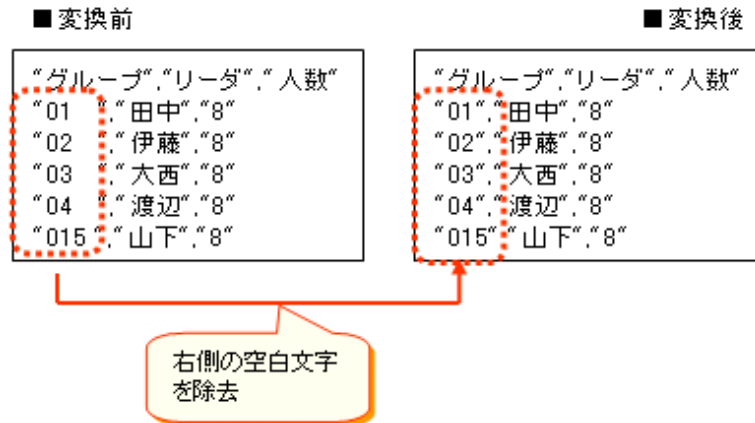
右詰の数字文字など、冒頭の文字を取り除いて数値として扱いたい場合などに利用できます。

なお、トリムできる項目の属性は、文字列型で、文字は、ASCII文字です。

#### 【例】ライトトリム

文字列の末尾にある空白を取り除く場合の例を示します。

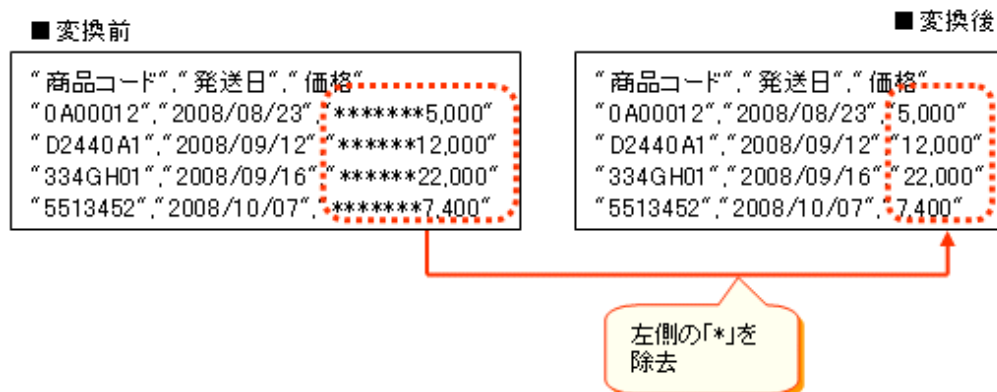
グループのそれぞれの番号の末尾に空白があり、以下の例では、番号の末尾に付いている空白を取り除いています。



#### 【例】レフトトリム

文字列の冒頭にある文字を取り除く場合の例を示します。

価格の冒頭に文字「\*」が付いており、以下の例では、文字「\*」を取り除いています。



### 1.4.3.6 パディング

文字列に0文字以上の文字を追加します。パディングでは、以下の2種類が利用できます。

- ・ ライトパディング

項目データの文字列の末尾に0文字以上の連続する任意の文字を追加します。これにより、指定した文字列長にします。

データベースヘデータを格納する際、char型属性項目の文字列末尾に、空白文字で埋め込む必要がある時などに利用できます。

- ・ レフトパディング

項目データの文字列の冒頭に0文字以上の連続する任意の文字を追加します。これにより、指定した文字列長にします。

右詰の数字文字への変換などに利用できます。

なお、パディングできる項目の属性は、文字列型で、文字は、ASCII文字です。

#### 【例】ライトパディング

文字列の末尾に空白を追加する場合の例を示します。

以下の例では、グループの番号の末尾に、空白を追加しています。

■ 変換前

```

"グループ","リーダー","人数"
"01","田中","8"
"02","伊藤","8"
"03","大西","8"
"04","渡辺","8"
"015","山下","8"

```

■ 変換後

```

"グループ","リーダー","人数"
"01","田中","8"
"02","伊藤","8"
"03","大西","8"
"04","渡辺","8"
"015","山下","8"

```

右側に空白文字  
を追加

【例】レフトパディング

文字列の冒頭に文字を追加する場合の例を示します。

以下の例では、従業員番号の冒頭に、文字「0」を追加して、6桁の番号に変換しています。

■ 変換前

```

"名前","従業員番号","年齢"
"田中","6345","51"
"上田","5077","28"
"木下","221","31"
"坂上","8392","31"

```

■ 変換後

```

"名前","従業員番号","年齢"
"田中","006345","51"
"上田","005077","28"
"木下","000221","31"
"坂上","008392","31"

```

「従業員番号」の  
冒頭に「0」を追加し、  
6桁の番号に変換

### 1.4.3.7 文字列の部分抽出

文字列に対して、文字の部分取得ができます。

【例】

文字列の部分抽出の例を示します。

以下の例では、郵便番号を、上3桁、下4桁に分割しています。

■ 変換前

```

"No","郵便番号","都道府県"
"01","222-0033","神奈川県"
"02","105-7123","東京都"
"03","211-8588","神奈川県"
"04","261-8588","千葉県"

```

■ 変換後

```

"No","郵便番号上3桁","郵便番号下4桁","都道府県"
"01","222","0033","神奈川県"
"02","105","7123","東京都"
"03","211","8588","神奈川県"
"04","261","8588","千葉県"

```

文字列の部分文字列  
を取得



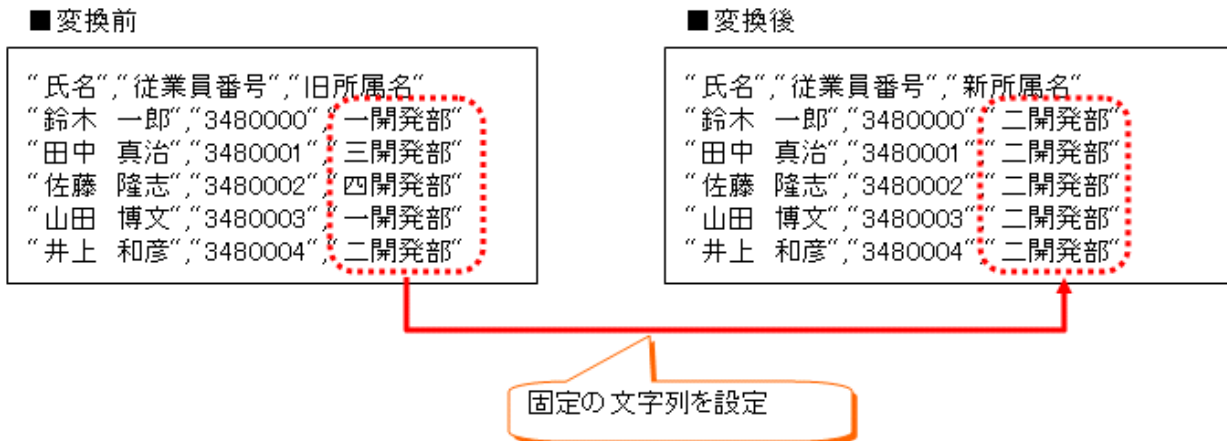
### 1.4.3.8 文字列の設定

固定の文字列の設定ができます。

以下に、文字列の設定の例を示します。

#### 【例】

文字列の設定の例を示します。以下の例では、所属を表わす文字列を「二開発部」と設定しています。



#### 注意

- NULL値、および半角空白のみの設定は、できません。
- 文字列の先頭に半角空白を設定しても、無効となります。文字列内、および文字列末尾の半角空白は有効です。

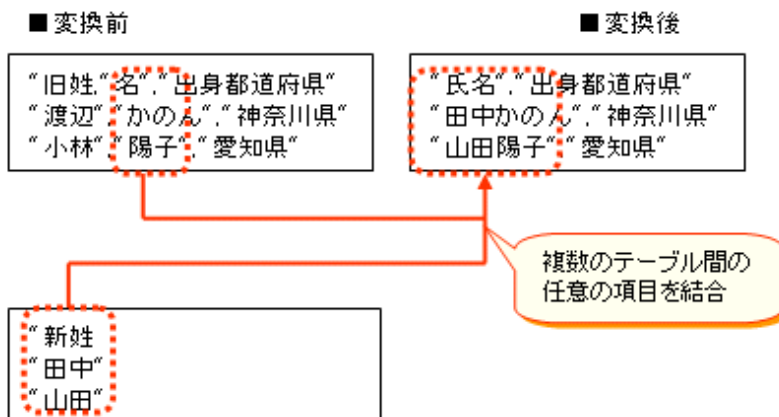
### 1.4.3.9 文字列の連結

文字列に対して、文字の連結取得ができます。

以下に、文字列の連結の例を示します。

#### 【例1】

以下の例では、別テーブルに存在する姓と名前を連結して氏名にしています。



#### 【例2】

以下の例では、氏名の前に所属を連結しています。

■ 変換前

"氏名","性別","出身都道府県"
"田中","1","東京都"
"鈴木","2","神奈川県"
"佐藤","1","大阪府"
"山田","1","福岡県"
"小林","2","愛知県"

■ 変換後

"氏名","性別","出身都道府県"
"第2 営業部)田中","1","東京都"
"第2 営業部)鈴木","2","神奈川県"
"第2 営業部)佐藤","1","大阪府"
"第2 営業部)山田","1","福岡県"
"第2 営業部)小林","2","愛知県"

文字列定数「第2 営業部」  
を連結

### 1.4.3.10 大文字/小文字変換

大文字から小文字、小文字から大文字へ変換できます。

【例】

項目「ローマ字読み」を大文字から小文字に変換します。

■ 変換前

"氏名","ローマ字読み"
"田中","TANAKA"
"山下","YAMASHITA"
"中田","NAKATA"

■ 変換後

"氏名","ローマ字読み"
"田中","tanaka"
"山下","yamashita"
"中田","nakata"

大文字→小文字

### 1.4.3.11 全角/半角変換

全角から半角、半角から全角へ変換できます。

【例】

項目「電話番号」を半角から全角に変換します。

■ 変換前

"電話番号","氏名"
"0333333333","田中"
"0525252525","山下"
"0123456789","中田"

■ 変換後

"電話番号","氏名"
0333333333"田中"
0525252525"山下"
0123456789"中田"

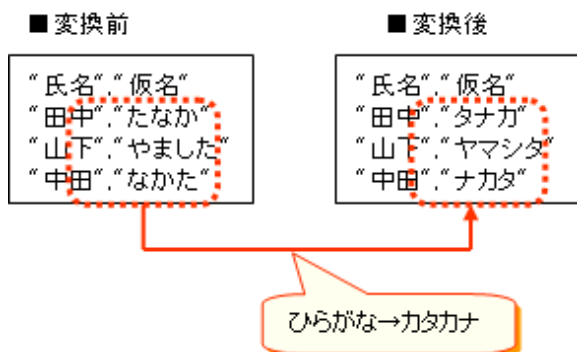
半角→全角

### 1.4.3.12 ひらがな/カタカナ変換

ひらがなからカタカナ、カタカナからひらがなへ変換できます。

#### 【例】

項目「仮名」をひらがなからカタカナに変換しています。



### 1.4.3.13 濁点/半濁点変換

全角仮名文字の濁点・半濁点を分離したり、結合したりします。

#### 【例】

分離する場合「かゝ」「は゜」、結合する場合「が」「ぱ」

### 1.4.3.14 四則演算

四則演算(加算、減算、乗算、除算)をすることができます。

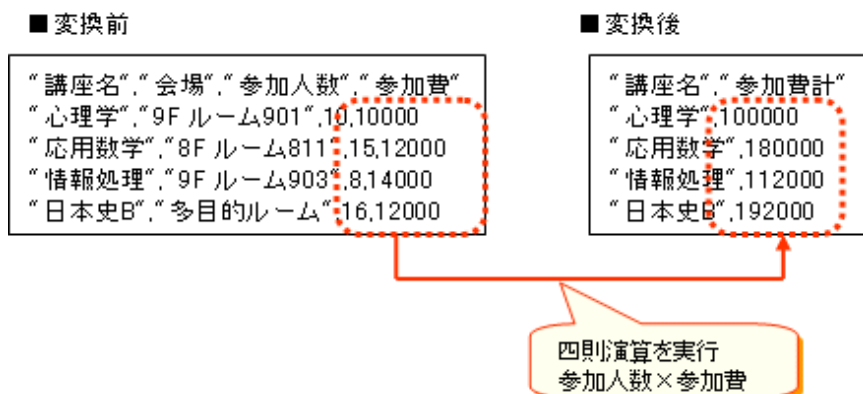
四則演算機能では、一つの数値属性項目、または複数の数値属性項目を利用して計算を行うことができます。同様に、出力項目は、数値属性となります。

利用できる演算子は、加算「+」、減算「-」、乗算「\*」、除算「/」の4種類であり、また、一般的な算術式と同様に、「( )」を利用することにより演算優先順位をつけることもできます。

演算は、収集したデータでも、一旦、何かの変換を実施した後のデータでも、演算できます。

#### 【例】

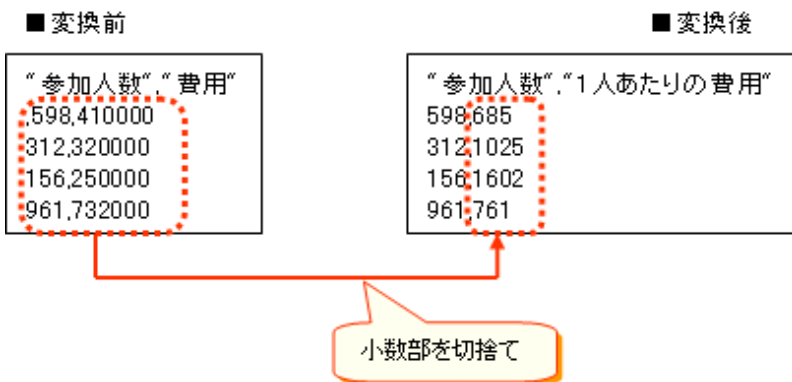
参加人数と参加費を掛けて、参加費計を出力する例を以下に示します。



演算結果の値を、切り捨てることができます。切り捨てる位置は、整数部、小数部で指定できます。

#### 【例】

1人あたりの参加費用を算出してから、小数部分を切り捨てる例を以下に示します。



### 注意

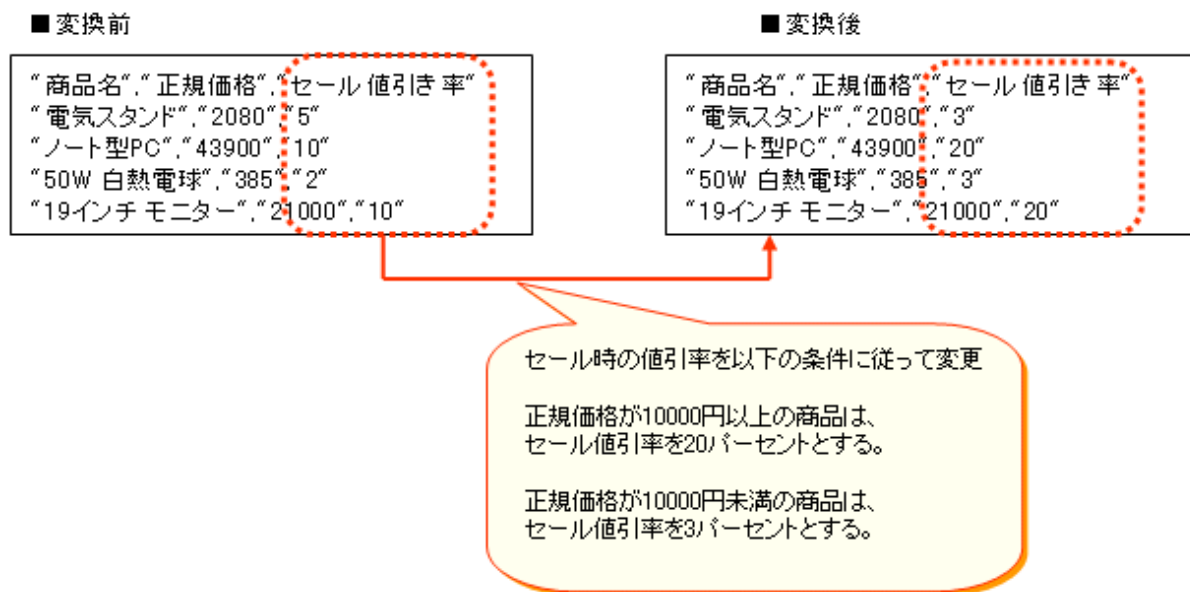
- 利用できる数値範囲は、18桁までの整数、18桁までの小数部を持つ最大36桁の値です。先頭の0を除く、整数部が18桁を超えるとエラーとなります。

### 1.4.3.15 項目条件判定

項目に条件を設定すると、条件に基づいて、固定値の設定やデータの演算処理をすることができます。

#### 【例】

商品の値引き率を条件に基づいて変更する例を示します。

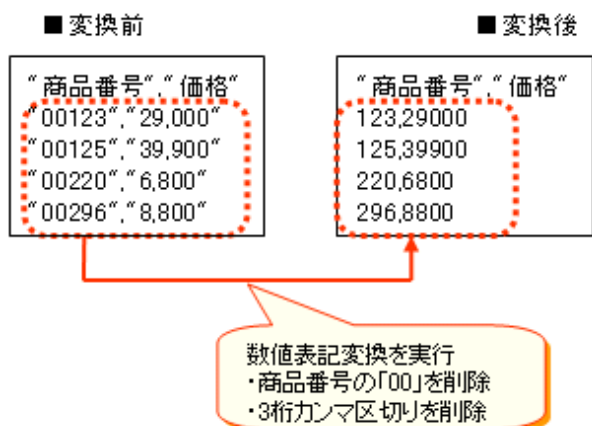


### 1.4.3.16 数字表記変換

ゼロ埋めの削除、3桁カンマ区切り(位取り)の削除し、文字属性から数値属性に変換します。

#### 【例】

5桁の文字列の商品番号のゼロ埋めを削除し、3桁の数値にします。また、3桁カンマ区切り表記の価格を、3桁カンマ区切りを取り、数値に変更する例を示します。



### 1.4.3.17 日時の書式変換

日時型属性を、「YYYY-MM-DD hh:mm:ss」表記に変換します。

日時の書式変換は、書式が異なる日付データの書式の統一に利用します。

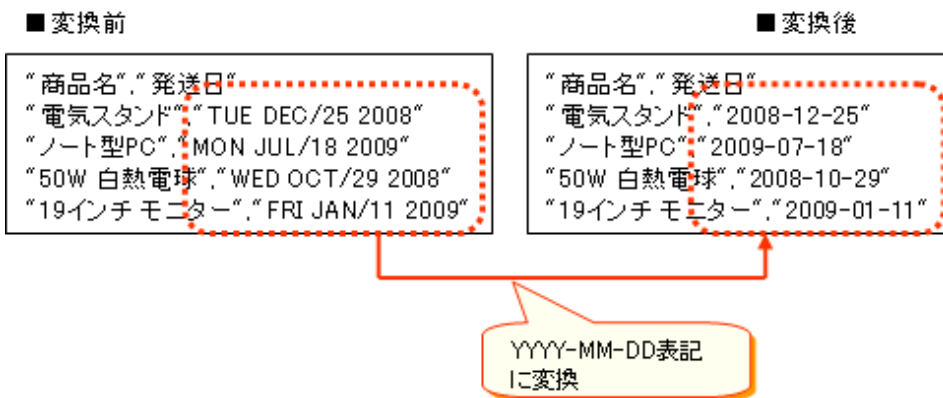
変換前のデータ形式と、変換後のデータ形式の対応を以下に示します。

表1.6 日時の書式の変換前のデータ形式と変換後のデータ形式の対応表

カテゴリ	変換前のデータ形式	Information Integratorで変換後のデータ形式
年	30～29 (意味: 1930～2029)	YYYY(年4桁)
	1～9999	
月	1～12	MM(月2桁)
	JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER	
	January, February, March, April, May, June, July, August, September, October, November, December	
	JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC	
	Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec	
日	1～31	DD(日2桁)
	SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY	
	Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday	
	SUN, MON, TUE, WED, THU, FRI, SAT	
曜日	SUN, MON, TUE, WED, THU, FRI, SAT	
	Sun, Mon, Tue, Wed, Thu, Fri, Sat	
	SUN, MON, TUE, WED, THU, FRI, SAT	
	Sun, Mon, Tue, Wed, Thu, Fri, Sat	
時	0～23	hh(時2桁)
分	0～59	mm(分2桁)
秒	0～59	ss(秒2桁)

#### 【例】

発送日の日付の書式を、「YYYY-MM-DD」に変換する場合の例を示します。



### 1.4.3.18 日時の部分取得

日時データから、「年」、「月」、「日」、「時」、「分」、「秒」のどれかを取得できます。

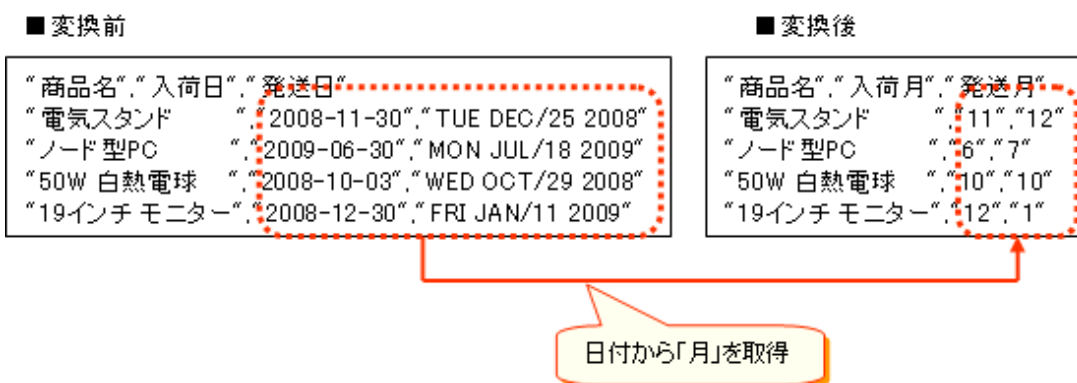
日時の部分取得は、以下の2パターンの取得方法があります。

- 日時データから、あるデータを取得  
例:「年」を取得する場合  
2009-07-10→2009
- 指定したデータに該当するデータを取得  
例:7月のデータを取得

取得前	取得後
2008-07-22	2008-07-22
2009-06-12	2009-07-11
2009-07-11	
2009-08-01	

#### 【例】

「YYYY-MM-DD」形式と、「DAY MON/DD YYYY」形式から、それぞれ「月」データを取得する場合の例を示します。



#### 注意

- 取得後の形式は、「YYYY-MM-DD hh:mm:ss」表記に変換します。詳細は、「1.4.3.17 日時の書式変換」を参照してください。
- 曜日は、部分取得できません。

- 一度に取得できるフィールドは、1つとなります。

### 1.4.3.19 項目編集時の注意事項

- データが存在しない場合は、変換結果をNULLとして扱います。
- 演算項目にNULLのデータが含まれている場合、演算結果は、NULLとなります。ただし、パラメタでNULLを0として扱うかどうかを選択できます。
- 書式を指定する場合、変換前のデータの内容が、データ書式と完全に一致している必要があります。指定したデータ書式と異なる場合は変換結果をNULLとして扱います。

## 1.4.4 レコード編集

### 1.4.4.1 振分け

振分け前のデータを振分け条件に従って分類します。

振分けできる条件には、以下の種類があります。

表1.7 振分けできる条件の種類

検索属性	検索条件	パターン検索の種類	パターン検索条件(※1)	内容
文字列検索	文字列の部分一致 (パターン検索)	文字列検索 (全角/半角文字列を 検索)	文字列一致	指定したキーワードを含む文書を抽出します。
			前方一致	指定したキーワードから始まる文書を抽出します。
			後方一致	指定したキーワードで終わる文書を抽出します。
			任意文字	指定した2つのキーワードが、任意の文字の前後に存在する文書を抽出できます。
			文字間隔	指定した2つのキーワードの間に、指定した文字数以内の文字が存在する文書を抽出できます。指定できる文字数は0～1024です。
			部分文字	指定した2つのキーワードの間に、指定した文字のどちらかを含む文書を抽出できます。
			文字範囲	指定したキーワードの一部の文字が、指定した範囲の文字を含む文書を抽出できます。
			数値範囲	指定したキーワードの一部の文字が、指定した範囲の数値である文字列のどちらかを含む文書を抽出できます。
		ワード検索 (英単語を検索)	ワード一致	指定した単語を含む文書を抽出できます。キーワードを連続して記述することで、連続した単語を含む文書を抽出できます。
			ワード間隔	指定した2つのキーワードの間に、指定した単語数以内の任意の単語が入ったフレーズを含む文書を抽出できます。
	文字列の完全一致	—	—	
	文字コードの大小	—	—	

検索属性	検索条件	パターン検索の種類	パターン検索条件(※1)	内容
数値検索	数値の一致	—	—	
	数値の大小	—	—	
日時検索	日付の一致	—	—	
	日付の大小	—	—	

※1:パターン検索条件は、論理演算子「&(論理積)、|(論理和)、~(否定)」で結合することにより複数の条件を組み合わせて抽出することができます。また、「( )」を指定することで優先順位もつけることができます。

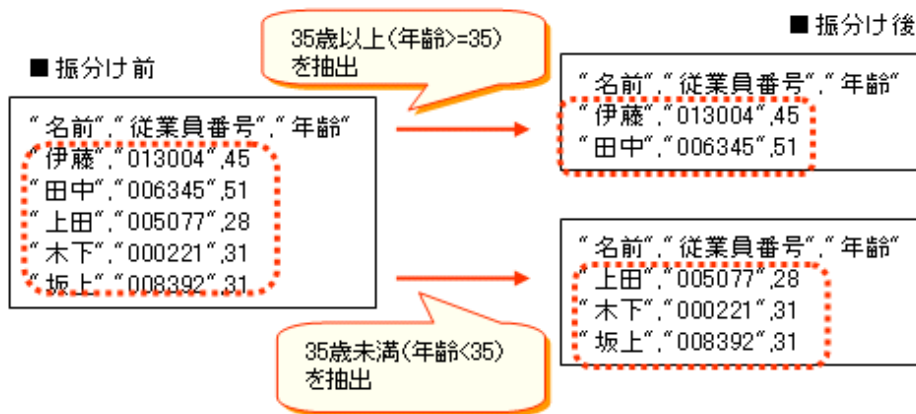
## ポイント

### 囲み文字の付与を変更する方法

振分け時に、データに対して囲み文字の付与または削除はできません。振分けデータに対して、囲み文字の付与または削除を行う場合は、対象のデータに対して変換処理(囲み文字の付与または削除)を実施してください。

### 【例1】数値検索

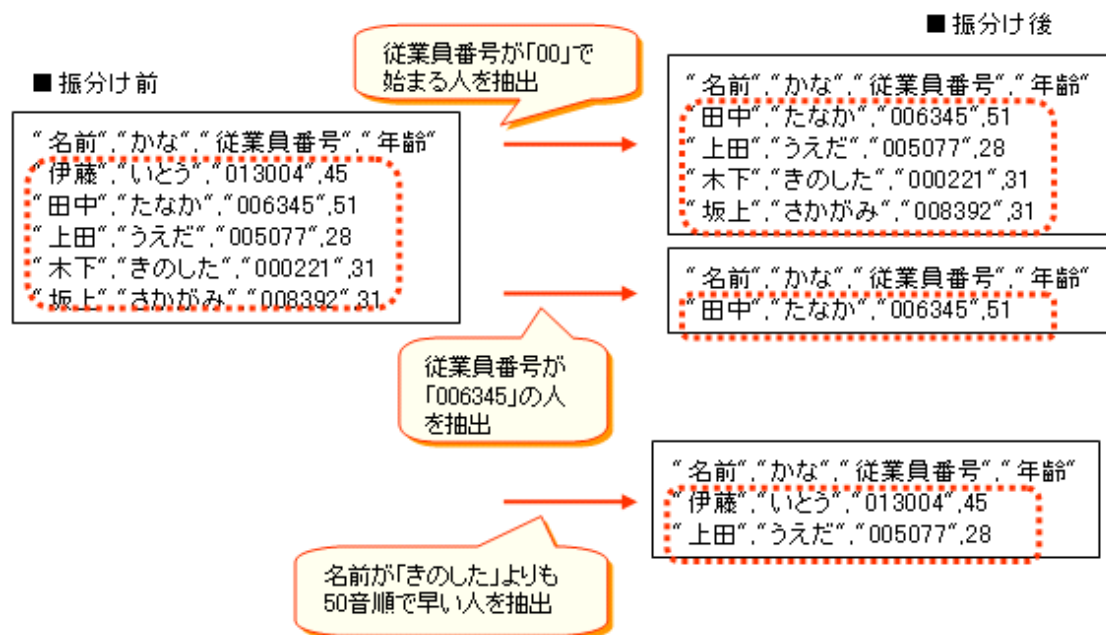
数値のキーワードとの大きさを比較して、条件を満たす文書を抽出する場合の例を示します。



### 【例2】文字列検索

文字列のキーワードとの大小の比較、完全一致、および部分一致する文書を抽出する場合の例を示します。

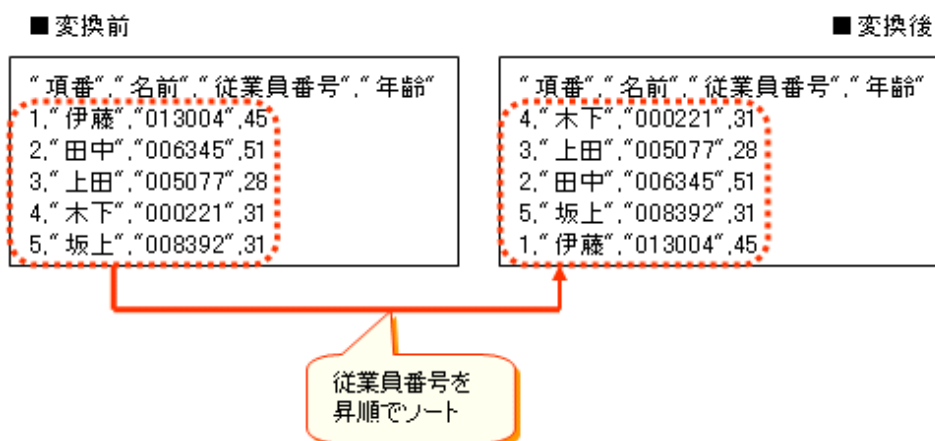




#### 1.4.4.2 ソート

任意の項目をソートキーに指定し、昇順/降順に並べ替えることができます。

また、ソートキーに指定した項目の属性によって、並べ替える方法が異なります。



#### 1.4.4.3 連結(ジョイン、ユニオン)

複数のデータを連結条件によって結合します。なお、一度に複数のマスタデータを連結することができます。最大128個のマスタデータと連結することができます。

連結には、以下の種類があります。

表1.8 連結の種類

機能名	結合型	結合条件	機能内容
ジョイン (横連結)	INNER JOIN	完全一致※1	結合条件が真のとき、項目を結合します。
		部分一致※2	偽のときは、レコードを削除します。
	OUTER JOIN	完全一致※1	結合条件が真のとき、項目を結合します。
		部分一致※2	偽のときもレコードを出力しますが、連結するマスタ項目は、NULLとなります。

機能名	結合型	結合条件	機能内容
ユニオン	UNION	完全一致	レコードを結合します。重複するレコードは削除されます。
(縦連結)	UNION ALL	—	レコードを結合します。重複するレコードも出力されます。

※1:完全一致は、連結キー同士が一致する場合

※2:部分一致は、マスタデータの連結キーがトランザクションデータの連結キーに含まれる場合。部分一致を複数満たすデータが存在する場合、出力データには、その数分だけ条件が真と判断され、結合します。

## ポイント

### ジョインの結合条件の最大設定数について

連結するマスタデータ1つ1つに対しての結合条件の最大設定数は、以下のとおりです。

- 結合条件が完全一致の場合  
連結するマスタデータ1つ1つに対して、それぞれ最大8個までの連結条件を指定することができます。
- 部分一致の場合  
連結するマスタデータ1つ1つに対して、それぞれ最大1個の連結条件を指定することができます。

それぞれ、以下で説明します。

## INNER JOIN

マスタデータの文字列が、トランザクションデータの文字列と完全に一致、または部分一致する場合に、項目を連結します。

INNER JOINの場合、トランザクションデータがマスタデータに存在しない場合、そのレコードを出力しません。

### 【例】

トランザクションデータの「商品No」と、マスタデータの「商品コード」をキーにして、項目を連結する場合の例を示します。(完全一致の場合)

#### ■ 変換前

・トランザクションデータ

日付	商品No	個数
"20090608"	"A03"	3
"20090609"	"A01"	24
"20090705"	"A04"	1
"20090707"	"A01"	60

トランザクションデータの値と、マスタデータの値が一致すれば、項目を連結

#### ■ 変換後

日付	個数	商品名	単価
"20090608"	3	"消しゴム"	80
"20090609"	24	"鉛筆"	60
"20090707"	60	"鉛筆"	60

商品コード	商品名	単価
"A01"	"鉛筆"	60
"A02"	"赤鉛筆"	100
"A03"	"消しゴム"	80

・マスタデータ

## OUTER JOIN

マスタデータの文字列が、トランザクションデータの文字列と完全に一致、または部分一致する場合に、項目を連結します。

OUTER JOINの場合、トランザクションデータが、マスタデータに存在しない場合、連結するマスタデータは、NULLとなります。

### 【例】

トランザクションデータの「商品No」と、マスタデータの「商品コード」をキーにして、項目を連結する場合の例を示します。(部分一致の場合)

■ 変換前

・トランザクションデータ

日付	商品No	個数
20090608	A03	3
20090609	H01	24
20090705	C04	1
20090707	D01	60

マスタデータの値が、トランザクションデータの値に含まれていれば、項目を連結

■ 変換後

日付	個数	商品名	単価
20090608	3	鉛筆	60
20090609	24		
20090705	1	消しゴム	80
20090707	60	定規	250

商品コード	商品名	単価
A	鉛筆	60
B	赤鉛筆	100
C	消しゴム	80
D	定規	250

・マスタデータ

ジョインの補足: 複数のマスタデータを用意する場合

マスタデータを2つ以上用意することで、3つ以上のジョインも可能となります。なお、この場合、1つのマスタデータにつき、1つ以上の連結条件が必要となります。

【例】

トランザクションデータの「商品No」と、マスタデータの「商品コード」、また、トランザクションデータの「日付」と、マスタデータの「日付」をキーにして、項目を連結する場合の例を示します。

■ 変換前

・トランザクションデータ

日付	商品No	個数
20090608	A03	3
20090609	A01	24
20090705	A04	1
20090707	A01	60

2つのマスタデータを用意して、連結

■ 変換後

日付	個数	商品名	単価	返品可能日
20090608	3	消しゴム	80	6/22
20090609	24	鉛筆	60	6/23
20090705	1			7/19
20090707	60	鉛筆	60	7/21

商品コード	商品名	単価
A01	鉛筆	60
A02	赤鉛筆	100
A03	消しゴム	80

・マスタデータ1

日付	返品可能日
20090608	6/22
20090609	6/23
20090705	7/19
20090707	7/21

・マスタデータ2

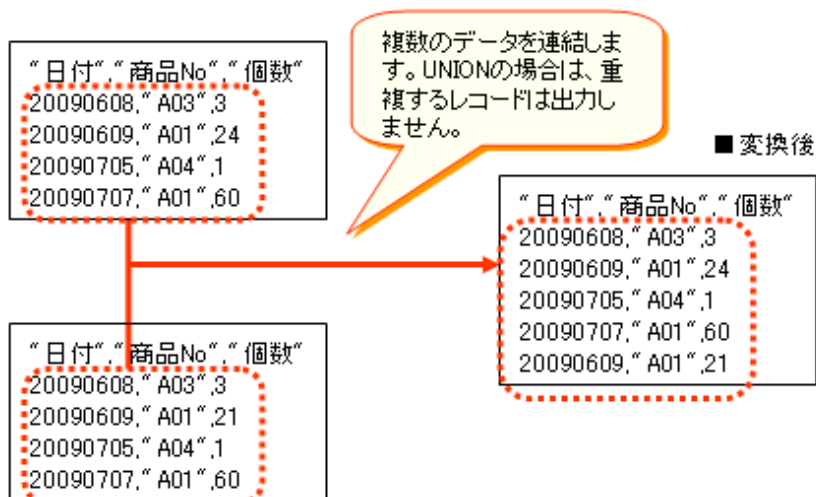
## UNION

複数のトランザクションデータのレコードを連結します。UNIONでは、重複したレコードは出力しません。

### 【例】

2つのトランザクションデータのレコードを連結する場合の例を示します。

#### ■ 変換前



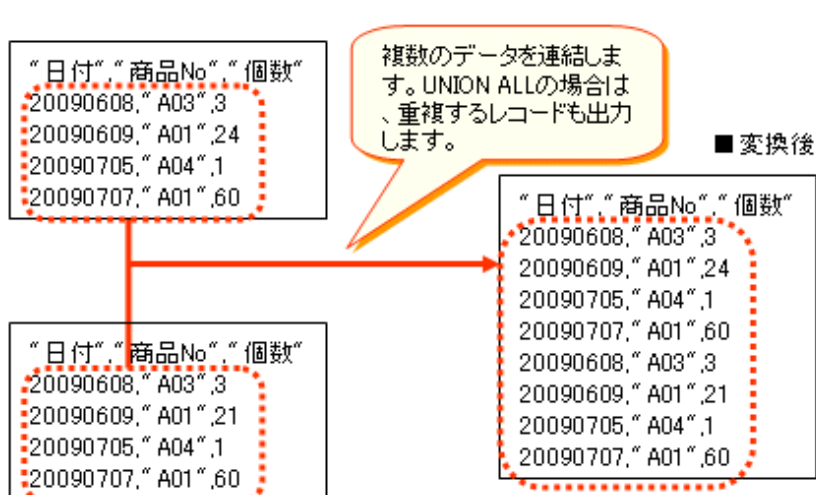
## UNION ALL

複数のトランザクションデータのレコードを連結します。UNION ALLでは、重複したレコードも出力します。

### 【例】

2つのトランザクションデータのレコードを連結する場合の例を示します。

#### ■ 変換前



### 1.4.4.4 集計

データをグループ化し、指定した項目の合計値を出力することができます。複数の項目をグループ化して、合計値を出力することもできます。

集計は、帳票に出力するときなどに利用することができます。

#### 【例1】

項目「クラス」単位で「数学」の値を集計する場合の例を示します。

このとき、集計対象以外の項目を出力させる場合、各グループの先頭レコードの値が表示されます。

### ■ 変換前

クラス	性別	氏名	数学
英語クラス	男	田中	76
英語クラス	女	青木	64
数学クラス	男	杉山	81
数学クラス	男	山田	75
数学クラス	女	鈴木	93

### ■ 変換後

クラス	性別	氏名	数学	合計点
英語クラス	男	田中	76	140
数学クラス	男	杉山	81	249
-	男	田中	76	389

全体の集計結果レコードを追加

項目「クラス」単位で集計

### 【例2】

集計する単位を複数指定する場合の例を示します。

指定された最初のキーでグループ化し、次に指定されたキーで、さらにグループ化します。

以下の例では、項目「クラス」でグループ化した後に、項目「性別」でグループ化し、各クラスの男女別の数学合計点を表示しています。

### ■ 変換前

クラス	性別	氏名	数学
英語クラス	男	田中	76
英語クラス	女	青木	64
数学クラス	男	杉山	81
数学クラス	男	山田	75
数学クラス	女	鈴木	93

### ■ 変換後

クラス	性別	氏名	数学
英語クラス	女	青木	64
英語クラス	男	田中	76
-	女	青木	140
数学クラス	女	鈴木	93
数学クラス	男	杉山	156
-	女	鈴木	249
-	-	青木	389

項目「クラス」と、項目「性別」で、「数学」を集計

## P ポイント

- ・ グループキーで指定した上位グループ単位の集計結果を任意に出力することができます。
- ・ 全体の集計結果のレコードを任意に出力することができます。
- ・ 上位グループや全体の集計結果を1レコード追加して出力することができます。このとき、グループキーの属性が文字型ならば半角文字の「ハイフン(-)」、数値型ならば「ゼロ(0)」が自動的に指定されます。

## 1.5 相手側ジョブ起動

### 1.5.1 相手側ジョブ起動

相手側ジョブ起動は、相手側システムのジョブを起動する機能です。ジョブの起動は、ユーティリティコマンドで実行できます。

## [相手側ジョブ起動の連携製品]

相手側システムのジョブを起動する場合、相手側システムの連携製品が相手側ジョブ起動をサポートしている必要があります。

Information Integratorがジョブ起動を要求できる相手側システムの連携製品は以下のとおりです。なお、相手側システムの連携製品がLinkexpress以外の場合、ジョブの実行結果を自システムが受け取ることはできません。

- Linkexpress
- Linkexpress File Transfer
- DTS
- HICS

Information Integratorに対してジョブ起動を要求できる相手側システムの連携製品は以下のとおりです。なお、相手側システムの連携製品がLinkexpress以外の場合、ジョブの実行結果を相手側システムに通知することはできません。

- Linkexpress
- Linkexpress File Transfer
- DTS
- HICS

備考.連携製品が相手側ジョブ起動をサポートしている通信プロトコルについては、連携製品のマニュアルを参照してください。

## [起動ジョブの実行資格]

起動ジョブの実行資格については、“[9.2 ファイル送受信、相手側ジョブ起動利用時のセキュリティ設定情報](#)”を参照してください。

## [相手側ジョブ起動の注意事項]

- 相手側ジョブが対話型のジョブの場合は、相手側ジョブ起動は利用できません。
- 通信プロトコルにSANおよびHULFTを使用する場合、相手側ジョブ起動は利用できません。

相手側システムのOSごとの注意事項は、以下のとおりです。

- 相手側システムがWindowsの場合
  - 相手側ジョブが以下のジョブの場合は、相手側ジョブ起動は利用できません。
    - ウィンドウが存在するジョブ(notepadなど)
    - ATコマンドの実行で正常動作しないジョブ
    - schtasksコマンドの実行で正常動作しないジョブ
    - Windows PowerShellで作成されたシェルスクリプト
  - 相手側ジョブ起動で実行されたプログラムからは、ネットワークドライブ、プリンタなどの資源にアクセスできません。
  - コマンド復帰タイミングとしてジョブ結果受信時を指定して相手側ジョブを起動する場合、このコマンドで起動するジョブは、Information Integratorに対してジョブ結果通知を行う必要があります。結果通知を行わないと、コマンドは復帰しません。また、ジョブ結果として0以外の値がジョブから通知された場合、コマンドは復帰値として1を返します。この場合は利用者がコマンドの出力メッセージからジョブの実行結果を判断してください。
  - 相手側ジョブ起動は、コマンドの復帰タイミングを以下のどちらかを選択して利用します。
    - 相手側ジョブのジョブ結果を受信したとき
    - 相手側ジョブを起動したとき

相手側ジョブのジョブ結果を受信したときに相手側ジョブ起動を復帰する場合は、相手側システムからInformation Integratorへジョブ結果を必ず通知してください。相手側システムからのInformation Integratorへのジョブ結果通知の方法については、Linkexpressのマニュアルを参照してください。

なお、相手側ジョブ起動の復帰タイミングは、拡張オペランド記述ファイル内の「jobrsp」オペランドで指定します。

- Windowsの仕様上、ジョブ起動に必要な利用者の権限は、Administratorsグループに所属している必要があります。Administratorsグループに所属していない場合、TR10325、TR20325、またはTR30325メッセージを出力してジョブの起動に失敗することがあります。また、ドメインを作成している場合、AdministratorsグループをDomain Adminsグループにする必要があります。なお、ジョブはジョブ起動時に指定したユーザーの権限で動作します。ただし、Windows Vista、またはWindows Server 2008上のジョブを起動する場合、起動されたジョブはシステム管理者権限で動作します。なお、相手側利用者は、拡張オペランド記述ファイル内の「prtsecur」オペランドで指定します。
  - 環境変数は、システムの環境変数が引き継がれます。path環境変数など、ジョブ独自の環境変数が必要な場合は、ジョブの内部で環境変数を変更してください。
  - 相手側システムがUNIXの場合
    - 相手側ジョブが以下のジョブの場合は、相手側ジョブ起動は利用できません。
      - 制御端末が必要なコマンド(passwdなど)
      - フルスクリーン系のコマンド(viなど)
    - 相手側ジョブ起動は、コマンドの復帰タイミングを以下のどちらかを選択して利用します。
      - 相手側ジョブのジョブ結果を受信したとき
      - 相手側ジョブを起動したとき

相手側ジョブのジョブ結果を受信したときに相手側ジョブ起動を復帰する場合は、相手側システムからInformation Integratorへジョブ結果を必ず通知してください。相手側システムからのInformation Integratorへのジョブ結果通知の方法については、Linkexpressのマニュアルを参照してください。

なお、相手側ジョブ起動の復帰タイミングは、拡張オペランド記述ファイル内の「jobrsp」オペランドで指定します。
  - 環境変数、umask値、標準出力、および標準エラー出力は、Information Integratorサーバを起動した環境(RCスクリプト、またはコンソール・端末)を引き継ぎます。相手側ジョブ起動に指定する利用者のシェル環境(.profile、.chsrc など)は引き継ぎません。また、ジョブの実行に必要な環境変数、およびジョブからのファイルアクセスに必要なumask値は、Information Integratorサーバを起動するRCシェルまたはコンソール・端末の実行環境に設定するか、またはジョブ内に設定してください。ジョブからの標準出力、および標準エラー出力は、Information Integratorサーバを起動したコンソール・端末に出力されます。Information Integratorサーバ起動後に端末を終了している場合は、出力がEIOのエラーとなる可能性があります。このため、ジョブからの標準出力、および標準エラー出力は、一時ファイルなどにリダイレクトするなどの考慮を行ってください。なお、コンソール・端末がロック・ハンクしている場合は、標準出力、および標準エラー処理で待ち状態になるため、ジョブが完了できなくなります。
  - UNIX上のジョブ起動の場合、ジョブはジョブ起動時に指定したユーザーの権限で起動します。ただし、ユーザーの二次グループに対するサポートは行いません。
- 相手側システムがグローバルサーバ/PRIMEFORCEの場合
  - 相手側ジョブ起動の復帰タイミングは、相手側ジョブを起動したときに行います。拡張オペランド記述ファイル内の「jobrsp」オペランドで本コマンドの復帰タイミングを指定します。必ず、起動時にコマンドが復帰するように「no」を指定してください。

## 1.5.2 ジョブ結果通知の有無

ジョブ結果通知とは、相手側ジョブ(注)が、Information Integratorサーバに対してジョブの実行結果を通知する処理のことです。なお、相手側ジョブ起動でジョブ結果通知を行う場合、FTP+、HTTP、またはHTTPSを使用する必要があります。

Linkexpress以外の連携製品と相手側ジョブ起動を行う場合、ジョブ結果通知を行わないように指定する必要があります。例えば、グローバルサーバ/PRIMEFORCE側のジョブを起動する場合は、ジョブ結果通知なしを指定してください。

注) 相手側ジョブ起動要求時、ジョブ結果通知ありが省略値の動作となります。

ジョブ結果通知の有無は、ifixejobコマンドの拡張オペランド記述ファイルで指定します。

### ジョブ結果通知あり

ジョブ結果通知を利用する場合、データソース定義を以下のように設定してください。

- 「プロトコル」に「FTP+」、「HTTP」、または「HTTPS」を指定します。
- ジョブ起動要求を行うシステムでは、「多重度-起動側」と「多重度-応答側」が1多重ずつ必要です。



- ・ ジョブを実行するシステムでは、「多重度-起動側」と「多重度-応答側」が1多重ずつ必要です。

ジョブ結果通知ありの場合、起動したジョブの中から必ずジョブ結果通知を発行する必要があります。ジョブ結果通知は、コマンドと利用者プログラムインタフェースを用意しています。

Information Integratorの場合: ifirspjobコマンド

Linkexpressの場合: lxrspjobコマンド

また、ジョブ結果通知未発行など、何らかの理由によりジョブ結果通知をInformation Integratorサーバへ通知できない場合を考慮して、ジョブ結果待ち時間を指定してください。ジョブ結果待ち時間は、起動側システムの要求時に指定できます。ジョブ結果待ち時間を省略した場合でジョブ結果通知未発行では、タイマ監視を行いませんのでジョブ起動要求が無応答状態になります。ジョブ起動時点から30分以内にジョブ結果通知が通知されない場合、メッセージ(TRx3014)が出力されます。ジョブ結果待ち時間が指定されている場合は、メッセージは、出力されません。

## ジョブ結果通知なし

ジョブ結果通知を利用しない場合、データソース定義を以下のように設定してください。

- ・ プロトコルに、「FTP」、「FTP+」、「HTTP」、「HTTPS」、または「HICS」を指定してください。
- ・ ジョブ起動要求を行うシステムでは、「多重度-起動側」が1多重必要です。
- ・ ジョブを実行するシステムでは、「多重度-応答側」が1多重必要です。



### 参考

グローバルサーバ/PRIMEFORCE上のジョブを相手側ジョブ起動で起動する場合、ジョブ結果通知なしを指定してください。

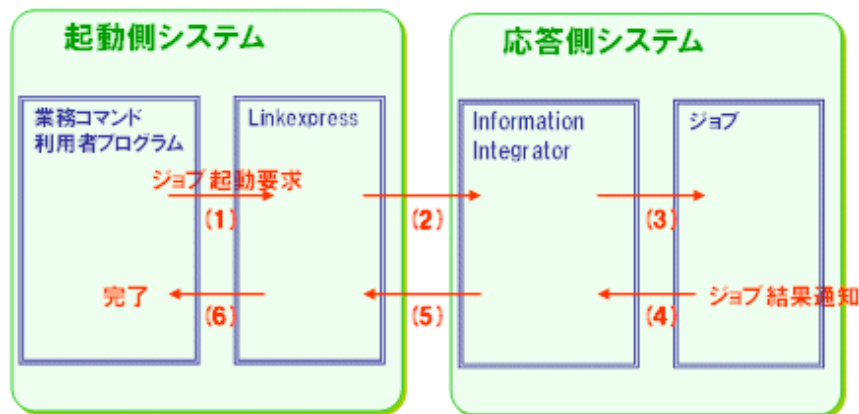
## 1.5.3 ジョブの起動インタフェース

相手側システムから起動されるジョブには、実行結果を通知するジョブと通知しないジョブがあります。

- ・ 実行結果を通知するジョブ

ジョブ結果通知を行うジョブは、実行完了時にInformation Integratorに対して結果通知を行います。ジョブの呼び出しシーケンスは以下のとおりです。

図1.26 実行結果を通知するジョブの処理の流れ



1. 利用者プログラム、ファイル転送コマンド、または業務からLinkexpressに対してジョブ起動要求を行います。
2. 起動側システムのLinkexpressから応答側システムのInformation Integratorに対してジョブ起動要求を行います。
3. 応答側システムのInformation Integratorがジョブを起動します。
4. ジョブから応答側システムのInformation Integratorに対してジョブ結果通知を行います。(注1)
5. 応答側システムのInformation Integratorから起動側システムのLinkexpressにジョブの実行結果が通知されます。
6. Linkexpressから利用者プログラム、ファイル転送コマンド、または業務にジョブの実行結果が通知されます。(注2)



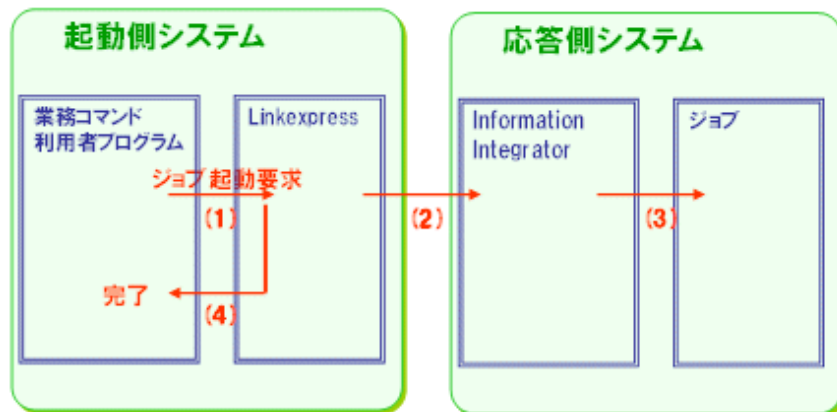
注1) (4)のジョブ結果通知が行われないか、異常終了した場合、(6)が通知されません

注2) (1)から(6)の処理のタイム監視は、利用者プログラム、ファイル転送コマンド、イベント定義、ネットワーク定義のjobtimeオペランドで指定可能です。

- ・ 実行結果を通知しないジョブ

ジョブ結果通知を行わないジョブは、ジョブの起動完了を実行結果として扱います。ジョブの呼び出しシーケンスは以下のとおりです。また、このタイプのジョブは、Linkexpress File Transfer、DTS、またはHICSから呼び出されることもあります。

図1.27 実行結果を通知しないジョブの処理の流れ



1. 利用者プログラム、ファイル転送コマンド、または業務からLinkexpressに対してジョブ起動要求を行います。
2. 起動側システムのLinkexpressから応答側システムのInformation Integratorに対してジョブ起動要求を行います。
3. 応答側システムのInformation Integratorがジョブを起動します。
4. (3)のジョブ起動後、Linkexpressから利用者プログラム、ファイル転送コマンド、または業務にジョブの起動完了が通知されません。

## 1.5.4 相手側ジョブ起動の動作環境

相手側ジョブ起動の動作環境は以下のとおりです。

### ID環境

Information Integratorが起動するジョブは、相手側ジョブ起動時に指定した利用者の権限で動作します。相手側ジョブ起動で使用する利用者情報の設定箇所については、“9.2 ファイル送受信、相手側ジョブ起動利用時のセキュリティ設定情報”を参照してください。

データソース定義の「利用者認証-logon」に「NO」を指定すると、Information Integratorを起動したシステム権限で動作します。

なお、Windows Server 2008上のジョブを起動する場合、上述の指定に関係なくシステム管理者権限で動作します。

### 環境変数

UNIX上のジョブを起動する場合、Information Integratorが起動したジョブの環境変数は、Information Integratorを起動した時の環境変数が引き継がれます。Windows上のジョブを起動する場合、システムの環境変数が引き継がれます。

環境変数「path」など、ジョブ独自の環境変数が必要な場合は、ジョブの内部で環境変数を設定してください。

また、Information Integratorを自動起動した場合の環境変数は以下のとおりです。

- ・ Windowsの場合、「SYSTEM」の環境変数が引き継がれます。
- ・ UNIXの場合、Information Integratorの起動シェルに設定した環境変数が引き継がれます。

### 動作環境の注意事項

UNIX上のジョブを起動する場合の注意事項です。

環境変数、umask値、標準出力、および標準エラー出力は、Information Integratorを起動した環境(RCスクリプト、またはコンソール・端末)を引き継ぎます。ジョブ起動に指定する利用者のシェル環境(.profile、.chsrc など)は引き継ぎません。

また、ジョブの実行に必要な環境変数、およびジョブからのファイルアクセスに必要なumask値は、Information Integratorを起動するRC

シェルまたはコンソール・端末の実行環境に設定するか、またはジョブ内に設定してください。

ジョブからの標準出力、および標準エラー出力は、Information Integratorを起動したコンソール・端末に出力されます。Information Integrator起動後に端末を終了している場合は、出力がEIOのエラーとなる可能性があります。このため、ジョブからの標準出力、および標準エラー出力は、一時ファイルなどにリダイレクトするなどの考慮を行ってください。

なお、コンソール・端末がロック・ハングしている場合は、標準出力、および標準エラー処理で待ち状態になるため、ジョブが完了できなくなりますので、注意してください。

## 1.5.5 ジョブの起動パラメタ

Information Integratorから起動されるジョブの起動パラメタは以下の形式で通知されます。起動パラメタのうち、ユーザー指定のジョブパラメタは起動元となるサーバがInformation Integrator、またはLinkexpress以外の場合も共通です。

### ジョブの起動形式

ジョブ結果通知を行う場合のジョブの起動形式は以下のとおりです。起動側システムから「ジョブ結果通知あり」を指定した相手側ジョブ起動の場合に、以下の形式でパラメタが通知されます。

```
ジョブ名 -r 相手側システム名 -i ジョブ識別番号 [ユーザー指定のジョブパラメタ]
```

ジョブ結果通知なしで相手側ジョブ起動を行う場合のジョブの起動形式は以下のとおりです。

```
ジョブ名 [ユーザー指定のジョブパラメタ]
```

#### 相手側システム名：

Information Integratorが使用するパラメタです。Information Integratorから起動されるジョブに通知されます。パラメタ種別として-rが使用されます。

#### ジョブ識別番号：

Information Integratorが使用するパラメタです。Information Integratorから起動されるジョブに通知されます。パラメタ種別として-iが使用されます。

#### ユーザー指定のジョブパラメタ：

ユーザーが必要に応じて使用するパラメタです。起動側システムが指定した利用者独自のパラメタは、上記のパラメタの後ろに追加されます。

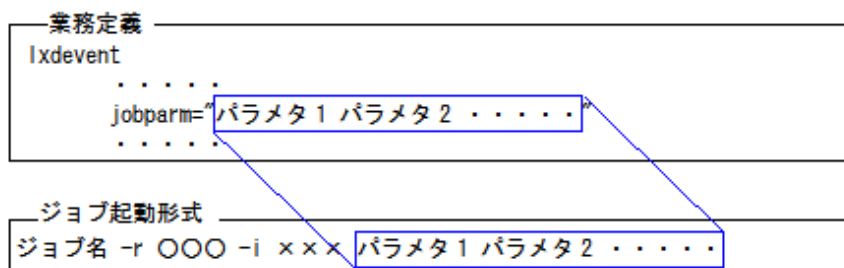
### ユーザー指定のジョブパラメタについて

ユーザー指定のジョブパラメタには以下の規定があります。これらの規定は、Information Integrator、またはLinkexpress以外のサーバと相手側ジョブ起動を行う場合でも共通です。

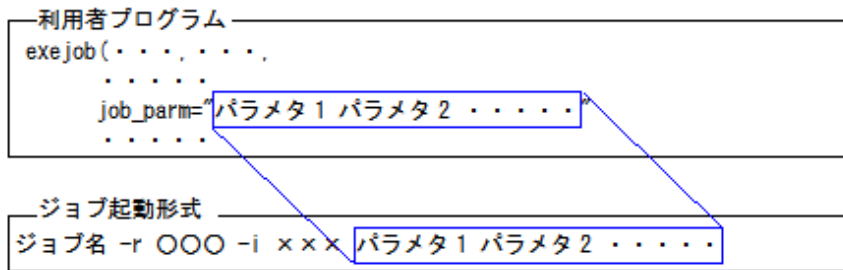
- 相手側ジョブ起動の相手側システムがHICSの場合、ユーザー指定のジョブパラメタは通知できません。
- ユーザー指定のジョブパラメタに空白文字を使うと、ジョブには複数個のパラメタとして扱われます。
- ユーザー指定のジョブパラメタには英数字を使用してください。

### 起動側システムがLinkexpressの場合

業務定義ではjobparmオペランドでユーザー指定のジョブパラメタを指定します。また、ファイル転送コマンドの場合はjobparamオペランドとなります。

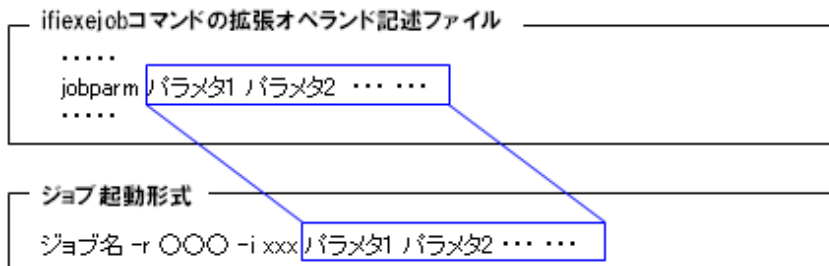


利用者プログラムではEXEJOBのjob\_parm、またはExecuteJobのJobParameterでユーザー指定のジョブパラメタを指定します。



### 起動側システムがInformation Integratorの場合

ifixejobコマンドの拡張オペランド記述ファイルのjobparmでユーザー指定のジョブパラメタを指定します。



Linkexpress File TransferやDTSなど、Linkexpress以外の連携製品からユーザー指定のジョブパラメタを指定する方法については、連携製品のマニュアルを参照してください。



起動側システムがグローバルサーバ/PRIMEFORCEのLinkexpress File transferまたはDTSの場合は、ジョブ起動パラメタを指定できません。

## 1.5.6 ジョブがInformation Integratorに行う情報通知

「ジョブ結果通知あり」でInformation Integratorから起動されたジョブは、以下の出力インタフェースで、Information Integratorにジョブの結果を通知する必要があります。なお、「ジョブ結果通知なし」で起動された相手側ジョブの場合、ジョブはInformation Integratorに対して通知を行う必要はありません。

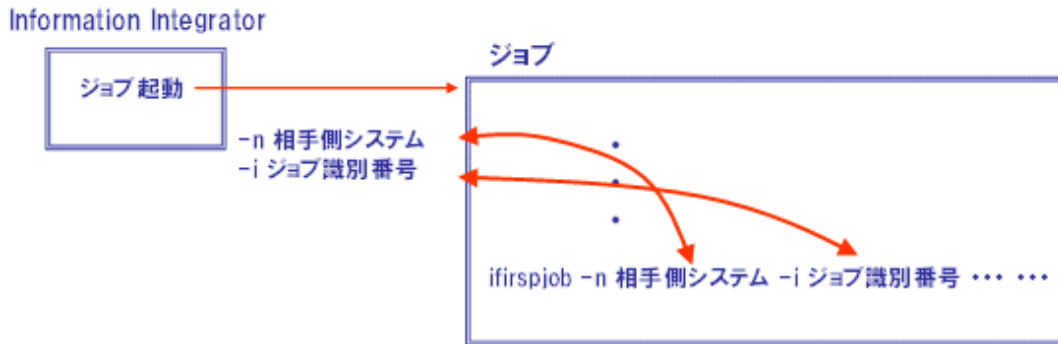
### ジョブ結果通知

ジョブ結果通知は、Information Integratorが起動したジョブが結果を通知するためのインタフェースです。ジョブ結果通知を行うことにより、Information Integratorは、ジョブの完了を認識します。なお、起動されたジョブが結果通知を行うには、データソース定義の「プロトコル」に、「FTP+」、「HTTP」、または「HTTPS」を指定する必要があります。

- Linkexpressの場合:lxrspbjobコマンド
- Information Integratorの場合:ifirspbjobコマンド

ジョブ結果通知で要求される設定値に、相手側システム名とジョブ識別番号があります。これらの設定値はジョブの起動時に入力インタフェースとして通知されます。

図1.28 ジョブ結果通知で要求される設定値



相手側システム: 「-n」オペランドとしてInformation Integratorから通知されます。

ジョブ識別番号: 「-i」オペランドとしてInformation Integratorから通知されます。

## 1.5.7 ジョブのサンプル

ジョブのサンプルについては、以下を参照してください。サンプルを利用する場合は、サンプルを任意のディレクトリ配下にコピー後、編集してください。

- Windowsの場合

```
Information Integratorサービンスツールディレクトリ¥sample¥RSPJOB. BAT
```

- UNIXの場合

```
/opt/FJSVifisv/sample/rspjob. sh
```

上記を利用する場合は、PATH環境の行のコメント(#)をはずしてください。また、必要に応じてパスを追加してください。

### ポイント

相手側システム名とジョブ識別番号は、それぞれ第2パラメタ(\$2)と第4パラメタ(\$4)で参照できます。

## 1.6 プロセス制御機能

### 1.6.1 処理プロセスの実行

Information Integratorでは、以下の3種類の実行方法があります。

- 利用者やアプリケーションがプロセスを実行する。
- スケジュール機能を利用して定期的にプロセスを実行する。
- イベントの発生を契機にプロセスを実行する。

収集元システムからデータが到着した場合など、イベントの発生を契機として処理が実行されるイベント監視実行です。対象となるイベントの発生を自動で検知して処理プロセスを実行することができるため、データの逐次処理を前提とした業務システムとの連携に利用できます。

それぞれ、以下の指定が可能です。

表1.9 スケジュール機能の種類

分類	種類	内容
スケジュール実行	日次	毎日、決まった時刻(時:分)に実行

分類	種類	内容
	週次	毎週、決まった曜日、時刻(時:分)に実行
	月次	毎月、決まった日付、時刻(時:分)に実行
	一定時間間隔繰り返し	指定された間隔で繰り返し実行 1~999(分)
イベント監視実行	イベント監視実行	収集元システムからデータが到着した場合など、イベントの発生を契機として、処理プロセスを自動的に実行  イベント監視実行についての詳細は、“ <a href="#">1.6.5 イベント監視</a> ”を参照してください。

## ポイント

一定時間間隔繰り返しと収集監視実行では、スケジュール実行時間帯を事前に設定しておくことができます。例えば、**Information Integrator**に、60分間隔で繰り返し実行するプロセスや、イベント監視実行のプロセスなどがあり、いずれのプロセスも実行開始時間帯を午前9時以降午後5時までと設定しておくことができます。

また、日次、週次などすべてのスケジュール機能では、プロセスの運用開始後のスケジュールの一時停止・再開を行うことができます。

## 1.6.2 自動リトライ

システム間のファイル送受信では、様々な原因で転送が異常となる場合があります。原因がネットワーク上の一時的な通信エラーでは、転送異常となったファイルの再送で問題が解決します。しかし、利用者による原因調査や再送のオペレーションなどの作業で再送までに時間が掛かることがあります。

自動リトライ機能では、転送異常の原因を自動判定し、ネットワーク上の一時的な通信エラーの場合はオペレーションなしで自動的に再送することで、利用者の負担軽減と再送までの時間短縮を支援します。

自動リトライは、ファイル送受信の要求単位にリトライし、指定したリトライ間隔とリトライ回数に従ってリトライします。リトライ回数分リトライしてもファイル集配信がエラー終了する場合、利用者には、最後のエラー情報が通知されます。

## 1.6.3 排他制御

多重実行時に各実行処理や、利用者プラグイン用アプリケーションの使用するリソースの関係で、多重実行ができない場合(DB抽出で同一データの二重抽出を抑制する場合や、プラグインとして利用する製品が多重実行できない仕様の場合など)に、**Information Integrator**内で排他できます。

これにより、同時実行によるデータ重複や過剰なエラー(多重度オーバ)といったリスクが軽減できます。

処理によって制御単位が異なります。以下の処理と制御単位の関係を示します。

表1.10 処理の種類と排他制御の関係

処理の種類	機能	制御単位	想定されるリスク	備考
データ収集	ファイル転送	相手システム名+データボックス	データ重複	運用種別が応答側の場合は無効
	DB抽出	DBMS種+データベース名+スキーマ名+テーブル名	データ重複 デッドロック	データ配付とも排他
	BLOB収集	エンドポイントURL+コンテナ名+プロブ名	データ重複	データ配付とも排他
データ配付	ファイル転送	相手システム名+データボックス	データ重複	運用種別が応答側の場合は無効
	DB格納	DBMS種+データベース名+スキーマ名+テーブル名	データ重複 デッドロック	データ収集とも排他

処理の種類	機能	制御単位	想定されるリスク	備考
	BLOB配付	エンドポイントURL+テナ名+プロポ名	データ重複	データ収集とも排他
データ変換	変換	データ変換ID	同一変換処理でのCPU占有	多重度制限の代替として制御
振分け	振分け	振分けID	同一振り分け処理でのCPU占有	多重度制限の代替として制御
プラグイン	プラグイン	プラグインID	データ重複	

データ収集でオブジェクト抽出機能を利用する場合およびデータ配付でオブジェクト格納機能を利用する場合は、排他制御機能はありません。

## 1.6.4 順序性制御

処理プロセスの実行中に同一の処理プロセスの実行を新たに行うときは、同一プロセスの多重処理の防止およびシステムの負荷軽減を行うことができます。

この順序性制御を行うことにより、先行の処理プロセスに関する処理が完了してから次の処理プロセスに関する処理を開始するように制御します。

## 1.6.5 イベント監視

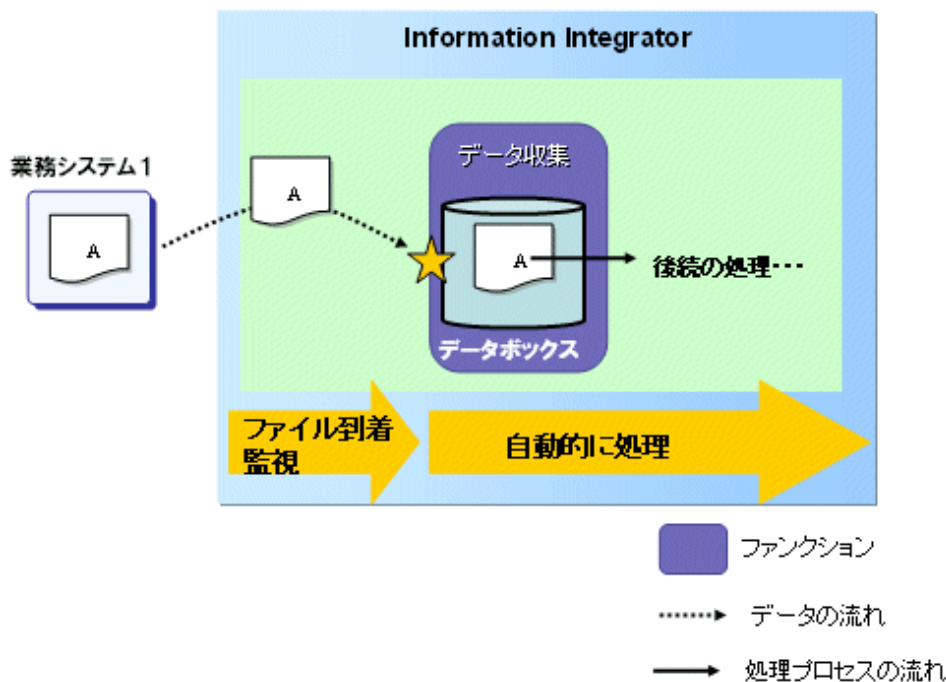
収集元システムからデータが到着した場合など、イベントの発生を契機として処理が自動的に実行される機能を、イベント監視と呼びます。

イベント監視では、以下のいずれかの契機で、処理プロセスを自動的に実行されます。

- ・ 他局主導でファイルを受信したとき

他局主導でファイルを受信したとき、ファイル転送の終了を契機に処理プロセスが自動的に実行されます。

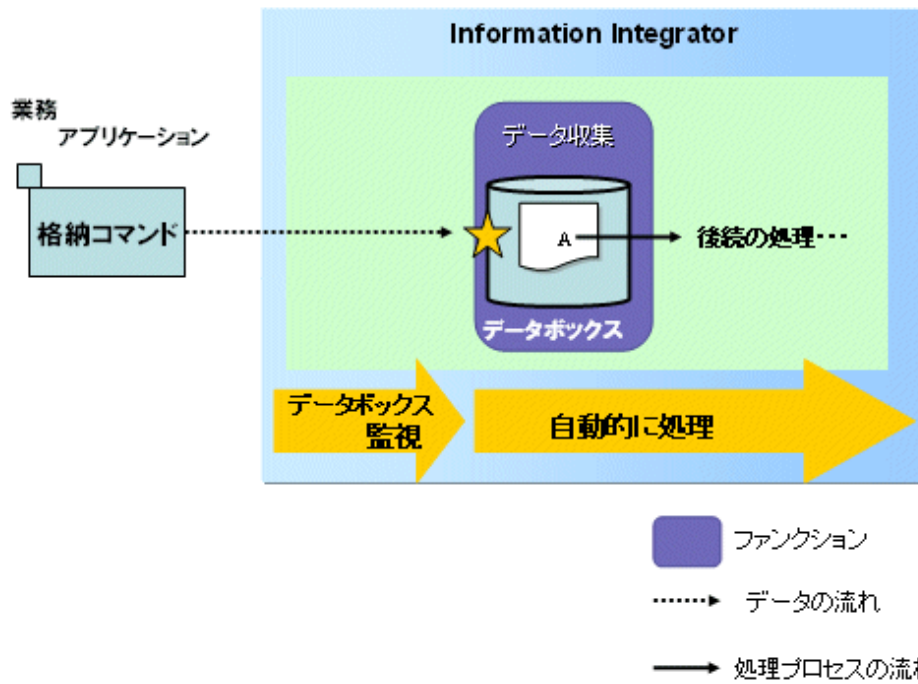
転送プロトコル(FTP+, HULFTなど)の種別には依存しません。





- データボックスにデータを格納したとき

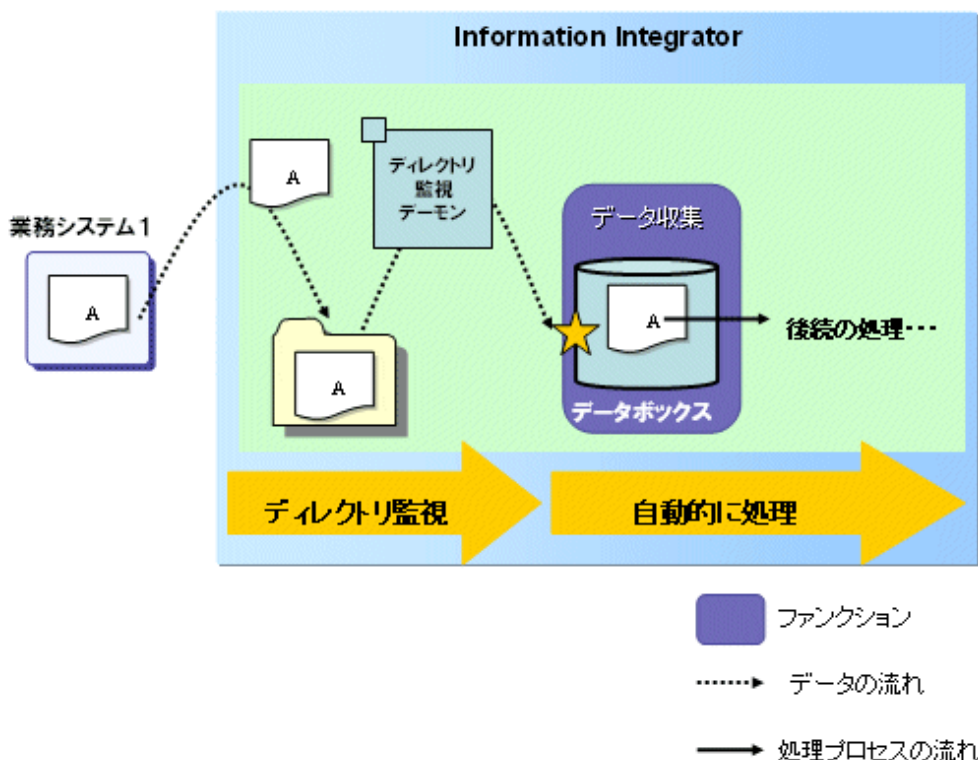
Information Integrator運用コマンド(「ifirstor」コマンド)を利用して、データボックスにデータを格納した契機に、処理プロセスが自動的に実行されます。



- 監視対象のディレクトリにファイルが生成されたとき

OSのコマンドや業務アプリケーションなどを利用して、処理プロセスの実行対象となるファイルを、あらかじめ監視対象に指定したディレクトリに生成します。監視対象のディレクトリにファイルが生成されたときに、処理プロセスが自動的に実行されます。

監視対象のディレクトリは、動作環境ファイルに指定します。詳細は、“セットアップガイド”を参照してください。



この場合の運用についての詳細は、“[1.6.5.1 ディレクトリを監視する場合](#)”を参照してください。

### 1.6.5.1 ディレクトリを監視する場合

処理プロセスの実行対象となるファイルが、他のアプリケーションで更新が完了したかどうか判断します。完了したときに、自動的に処理プロセスが実行されます。

大量データでは、書き込みの開始から完了までに時間を要します。そのため、更新日付が現在時刻以前である場合、かつ、ファイルに排他が掛かっていない場合に、書き込みが完了したものと判断します。

監視対象のディレクトリに生成された処理対象ファイルは、処理プロセスが実行されると、自動的に削除されます。

以下に、利用方法について説明します。



#### 注意

通常は、監視対象のディレクトリには、処理プロセスの実行対象となるファイルだけが生成されるようにしてください。

監視対象のディレクトリに生成されたファイルのうち、処理プロセスの実行開始時に処理プロセスの実行対象となるファイルを検索します。この検索に時間を要するためです。

また、ファイルの更新完了は前述の手段で判断していますが、以下のいずれかの方法でファイルを生成するようにしてください。特に監視対象のディレクトリにデータ量の大きいファイルを生成する場合には必ず以下のいずれかの方法で生成してください。

- 別のディレクトリに一度作成してから、監視対象のディレクトリに配置する。
- ファイルの書き込み中には、排他をかける。

#### 処理プロセスの実行対象となるファイルを指定

監視対象のディレクトリに生成されたファイルのうち、データ収集定義の収集ファイル名に処理プロセスの実行対象となるファイル名をあらかじめ指定します。

- 完全一致  
ファイル名が完全に一致するファイルのみを対象とする場合
- ワイルドカード指定  
前方一致、後方一致など、ファイル名の一部分が一致するファイルを対象とする場合

詳細は、“[III Studio ヘルプ](#)”または、“[デザインシート ヘルプ](#)”を参照してください。

#### 処理プロセスの実行対象外となるファイルの扱い

監視ディレクトリに生成されたファイルのうち、処理プロセスの実行対象外となるファイルの扱いをあらかじめ指定しておくことができます。

対象外のファイルの扱いは、以下のとおりです。

- 対象外ファイルを自動的に削除する
- 対象外ファイルを別のディレクトリに退避する

これらは、動作環境ファイルに指定します。詳細は、“[セットアップガイド](#)”を参照してください。



#### 注意

監視ディレクトリ配下に、処理プロセスの実行に不要なファイルを配置し続けないようにしてください。



#### ポイント

対象外ファイルを別のディレクトリに退避する場合



退避後のファイル名は、元のファイル名(監視対象ディレクトリに生成されたファイル名)に、日付と通番が付与されます。これにより、同一ファイル名のファイルが複数あっても、区別して管理することができます。

---

### 1.6.5.2 複数のイベントを監視する場合

複数のイベントを監視すると、以下のどちらかの契機で処理プロセスを自動的に実行することができます。

- すべてのイベントにおいて、ファイルが到着したとき  
処理プロセスの実行対象となるファイルがすべて到着するまで待ち合わせます。
- いずれかのファイルが到着したとき  
複数のイベントを監視し、いずれかのイベントにファイルが到着したときに、自動的に処理プロセスが実行されます。すべてのファイルを待ち合わせません。

なお、複数のイベントを監視する場合の実行契機は、データ収集定義の「0件データの扱い」でカスタマイズすることができます。詳細は、“[6.6.3 収集方法の設計](#)”(III Studioを利用して設計する場合)または、“[7.4.3 収集方法の設計](#)”(デザインシートを利用して設計する場合)を参照してください。

以下に例を示します。

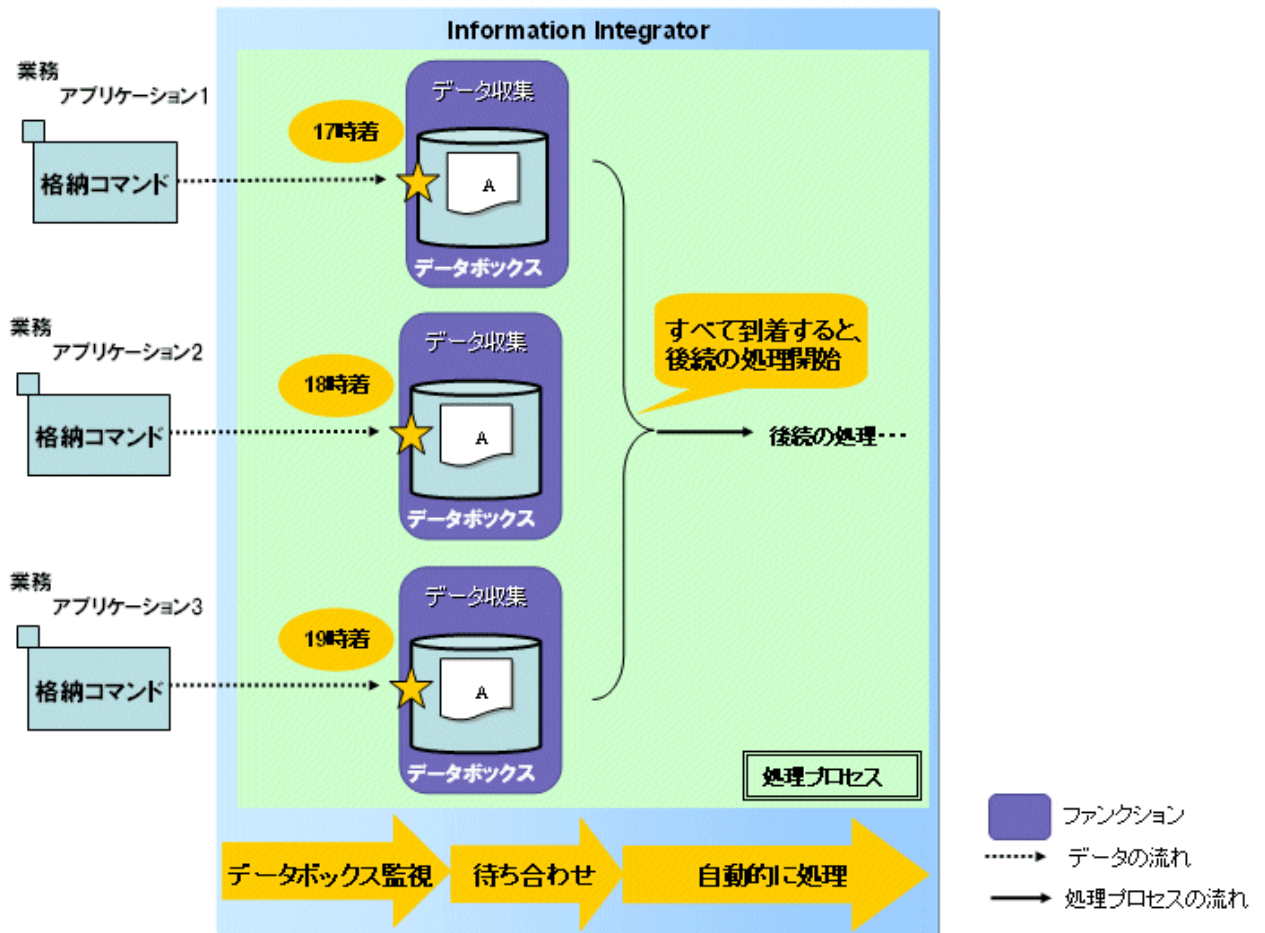
#### ポイント

---

以下の例は、すべてのイベントがデータボックスに格納するイベント形態ですが、他局主導でのファイル受信したときや、監視対象のディレクトリにファイルが生成されたときを契機としたイベントとの混在も可能です。

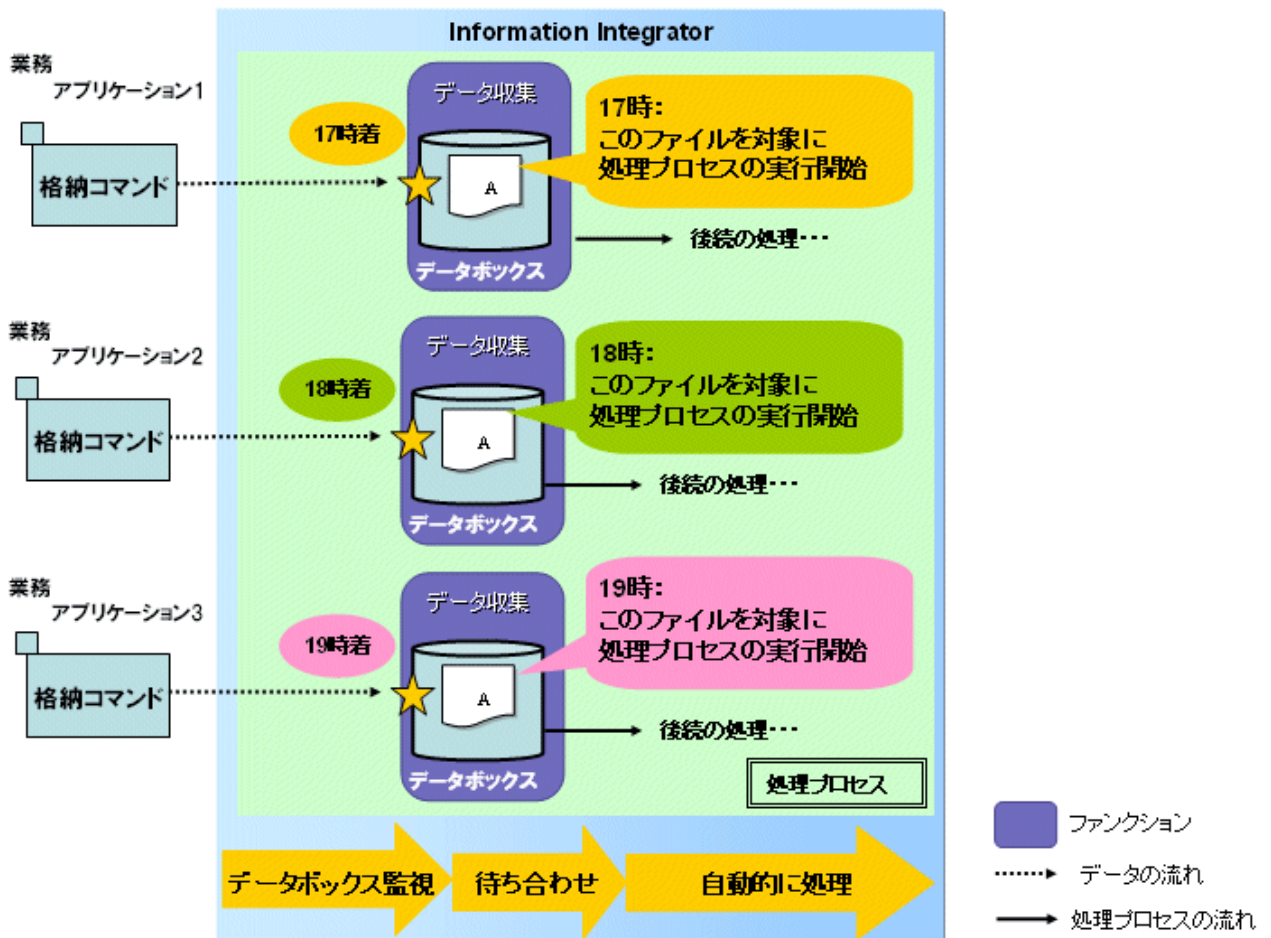
---

- 例)すべてのイベントにおいて、ファイルが到着するまで待ち合わせる場合  
業務アプリケーション1～業務アプリケーション3より転送されるファイルがすべて到着するまで待ち合わせます。



- 例)いずれかのファイルが到着したとき、処理プロセスが自動的に実行される場合

1日一回定刻にデータボックスにデータを格納する場合、17時、18時、19時の計3回、データボックスに格納されたファイルを対象に、処理プロセスが実行されます。



### 1.6.5.3 イベントの監視に休止期間を設ける・一時停止する

業務時間外でのイベントの発生時などに備え、あらかじめイベント監視に休止期間を設けたり、クラスタ運用切り替え後のリカバリ操作中のイベントの発生などに備えて、イベント監視を一時停止したりすることができます。

- イベント監視に休止期間を設ける

日々の業務時間が明確である場合、イベント監視時間帯をあらかじめ指定します。動作環境ファイルの「sch\_workingtime」に指定します。詳細は、“セットアップガイド”を参照してください。

- イベント監視を一時停止する

イベント監視を一時的に停止する場合は、処理プロセスのスケジュールを一時停止します。

再開する場合は、処理プロセスのスケジュールを再開します。

スケジュールの一時停止およびスケジュールの再開は、運用管理クライアントまたは、運用コマンドで行います。詳細は、“運用ガイド”を参照してください。

なお、処理実行中に休止時間に入った場合や処理プロセスの実行を一時停止した場合は、そのまま処理は継続されます。

また、休止時間中や一時停止中にイベントが発生した場合、処理プロセスは自動的に実行されません。イベント監視時間帯に入った後または、処理プロセス実行を再開した後に処理プロセスを実行してください。処理プロセスの実行は、運用管理クライアントまたは、「ifixeprc」コマンドを利用して行います。

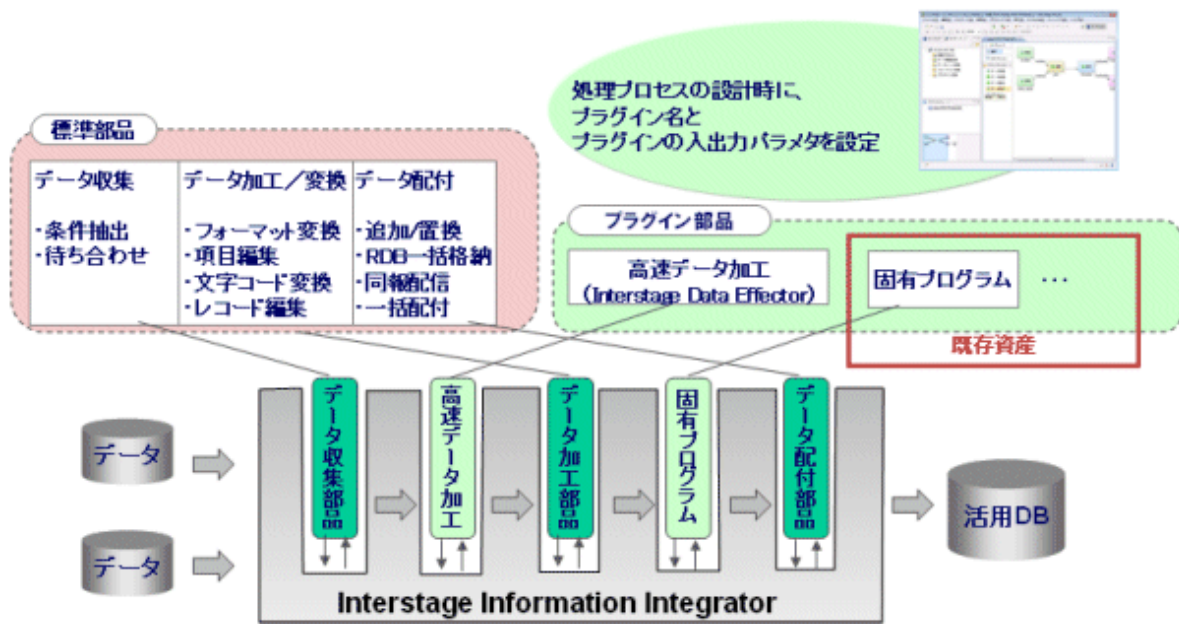
## 1.7 プラグイン

既存システムに使用していた変換アプリケーションや、各種拡張ツールをプラグイン機能により、Information Integratorに組み込むことができます。

プラグイン機能では、収集や変換、配付といった処理の単位でプラグインを組み込む機能(以降、処理プラグイン機能と呼びます。)と、変換処理時に項目単位に必要な変換アプリケーションをプラグインとして組み込む機能(以降、変換プラグイン機能と呼びます。)があります。

それぞれのプラグイン機能について説明します。

### 1.7.1 処理プラグイン機能のイメージ



組み込むことができるアプリケーションは、以下のとおりです。

表1.11 プラグインで利用可能なアプリケーション

プラットフォーム	利用可能なアプリケーション
Windows	exe/bat形式(実行可能形式)。 コマンドプロンプトから実行可能なバイナリ・プログラム、およびスクリプト(バッチファイル形式)
UNIX	コンソールより実行可能なバイナリ・プログラム、および各種スクリプト

#### 注意

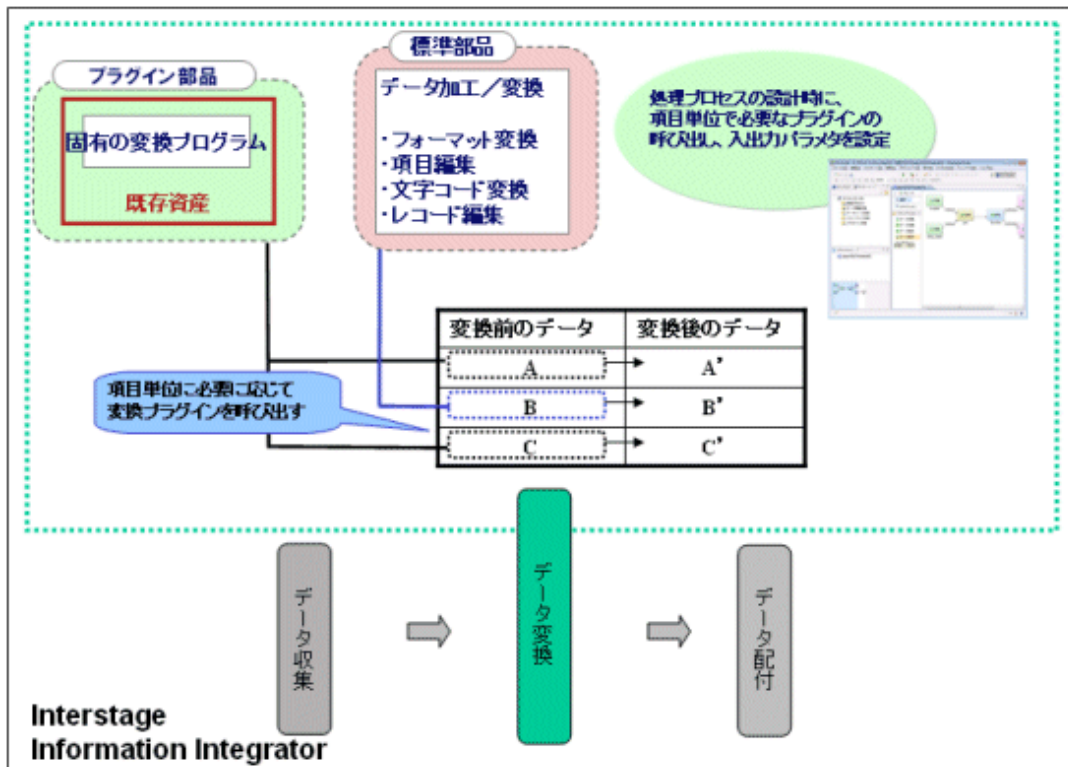
以下のプログラムについては実行できません。

- Windowsの場合
  - ウィンドウが存在するプログラムまたはコマンドなど(notepadなど)
  - ATコマンドの実行で正常動作しないプログラムまたはコマンドなど
  - schtasksコマンドの実行で正常動作しないプログラムまたはコマンドなど
  - Windows PowerShellで作成されたシェル・スクリプト

- UNIXの場合
  - ー 制御端末が必要なコマンド(passwdなど)
  - ー フルスクリーン系のコマンド(viなど)

処理プラグイン機能で組み込み可能なアプリケーションの詳細は、“[H.1 処理プラグイン機能で組み込み可能なアプリケーション](#)”を参照してください。

## 1.7.2 変換プラグイン機能のイメージ



組み込むことができるアプリケーションは、以下のとおりです。

### プラグインで利用可能なアプリケーション

利用可能なアプリケーションは、C言語またはCOBOL言語で作成され、Information Integratorのアドレス空間内にマッピング可能な動的ライブラリに含まれる形で提供されたサブルーチン(関数やAPIなど)です。

### 注意

日付属性/日付時間属性のデータを扱う場合のデータ形式について

変換プラグイン実行後の形式は、以下のとおりです。

- 日付属性の場合:「YYYY-MM-DD」
- 日付時間属性の場合:「YYYY-MM-DD hh:mm:ss」

実行できないプログラムについて

以下のプログラムについては実行できません。



- アプリケーションの呼び出し1回につき、複数項目の処理結果を返却するような処理はできません。アプリケーションの呼び出し1回につき、処理結果の返却項目は1項目だけです。  
ただし、アプリケーションの呼び出し1回につき、複数項目のデータをアプリケーションに通知することは可能です。
- オブジェクト指向型言語で多用される、getter/setterのように事前に実行手順が必要なアプリケーションは利用できません。個々の呼び出しだけで処理が完結されるアプリケーションを指定してください。
- 標準ライブラリに属する関数を変換プラグインに指定することはできません。

変換プラグイン機能で組み込み可能なアプリケーションの詳細は、“[H.2 変換プラグイン機能で組み込み可能なアプリケーション](#)”を参照してください。

## 変換プラグインと連携可能なデータ形式

CSVファイル形式またはバイナリ形式のデータと連携可能です。

# 1.8 管理機能

## 1.8.1 定義作成/定義管理

Information Integratorでは、処理プロセスの設計情報をシステムの定義情報として、そのまま実行環境で利用できます。

### 定義作成

処理プロセスの処理手順および処理プロセスを実行するために必要な各種パラメタの設計情報に関する定義を行います。

これらは、III Studioまたは、デザインシートを利用して行います。

- III Studio

各定義を視覚的に確認しながら設計を行うツールを「III Studio」と呼びます。

III Studioは、GUIベースの統合開発環境Interstage Studio上で動作するプラグインです。統合開発環境の統一された操作性が特長です。

- デザインシート

各定義をExcelファイルに入力して設計を行うツールを「デザインシート」と呼びます。Excelの操作で容易に入力が可能です。

### ポイント

#### 作成効率を向上させる「メタ収集機能」

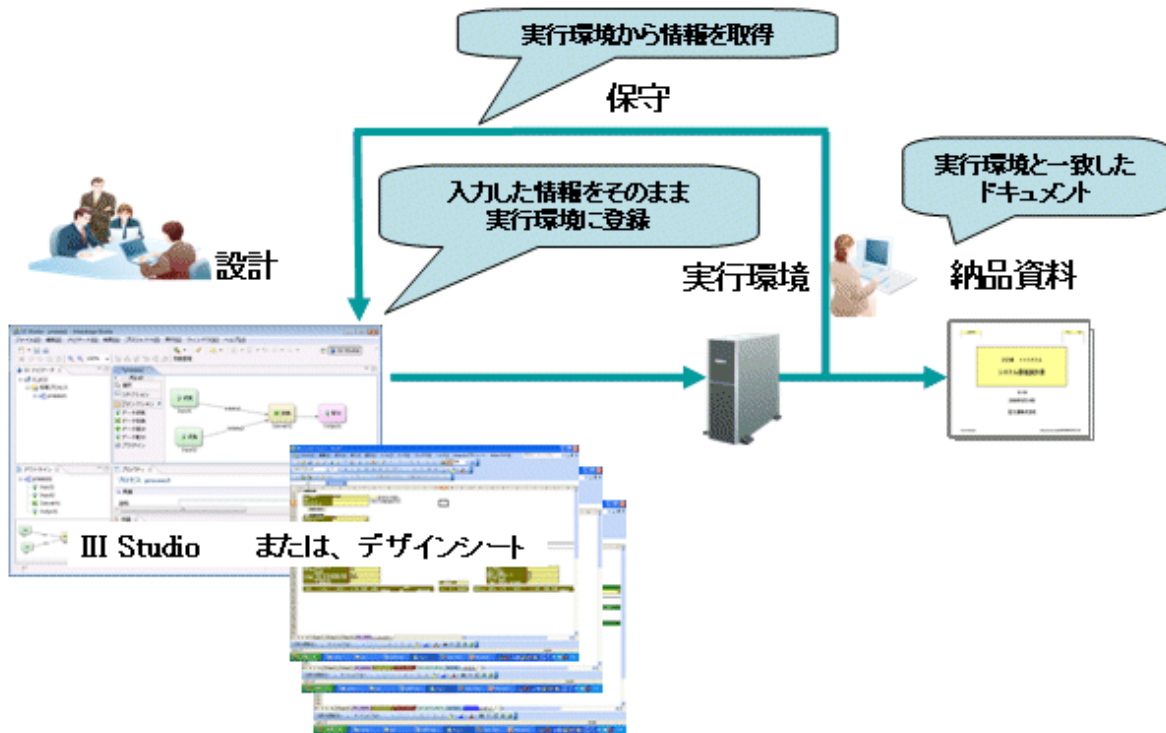
収集元システムや配付先システムにある、データの形式や処理方法に関する情報をメタデータと呼びます。Information Integratorでは、データ構造(項目名やデータ属性など)のメタデータを取り込んで設計を行うことができます。

本機能により、III Studioやデザインシートへの情報入力時に、メタデータを最初から作成する手間がなくなり、定義作成効率が大幅に向上します。

メタ収集機能についての詳細は、“[1.8.1.1 メタ収集機能](#)”を参照してください。

III Studioまたはデザインシートで設計した情報を実行環境に登録して利用します。設計から保守にわたる一連のシステム開発フェーズは、III Studioまたは、デザインシートを起点に行われます。

図1.29 Information Integratorの定義作成のイメージ



## 定義管理

実行環境に登録された定義を取り出したり、参照したりする一連の定義情報は、III Studioと「定義管理クライアント」で管理されます。定義の参照、修正は、以下のいずれかの方法で行います。

- III Studio
- デザインシート

デザインシートで定義の参照、修正を行う場合は、「定義管理クライアント」を利用します。「定義管理クライアント」は、定義情報を管理するツールです。

### 1.8.1.1 メタ収集機能

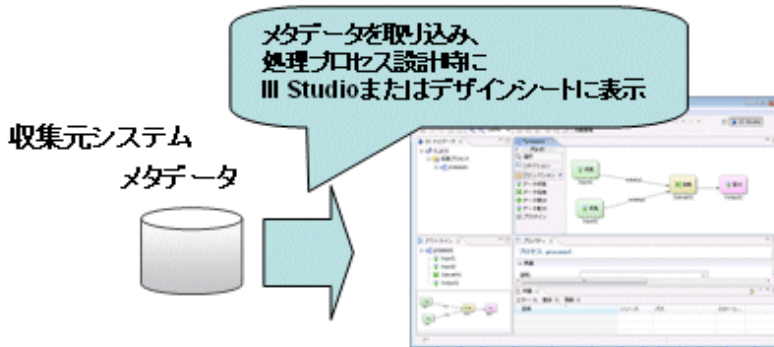
収集元システムや配付先システムにあるメタデータをInformation Integratorに取り込みます。

Information Integratorでデータ変換・編集を行う場合、あらかじめ変換前と変換後のデータ構造に関する情報を入力します。この定義を行うとき、変換対象となるデータのメタデータを取り込む機能です。

以下のデータのデータ構造に関するメタデータを収集することができます。

- CSVファイル
- COPY LIBファイル  
COBOLのデータ項目に関する定義(COBOL COPY句)が記述されたファイルです。
- Symfoware Server
- Oracle
- SQL Server
- SQL Azure
- Salesforceオブジェクト
- 帳票データ

収集できるメタデータは、収集が可能なデータの形式や、データ型などの条件に準じます。条件の詳細については、“[付録B Information Integratorで扱えるデータ形式について](#)”を参照してください。

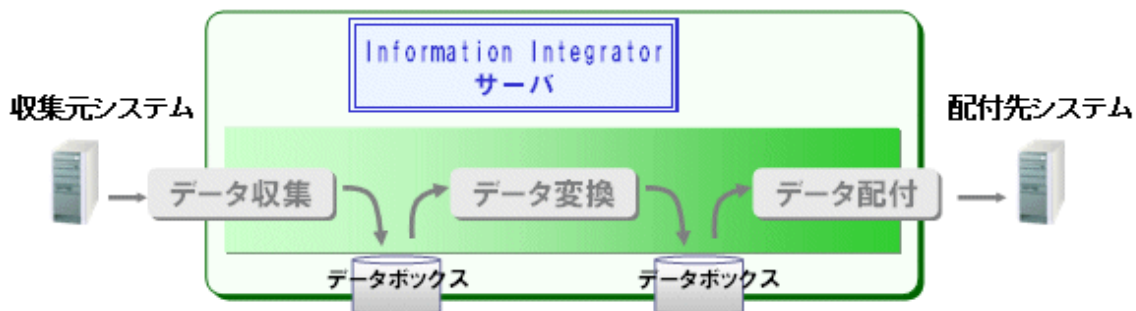


## 1.8.2 データ管理

Information Integratorでは、収集元システムから収集したデータや、配付先システムに配付したデータなどをInformation Integratorサーバ内に保持します。この保持する領域を「データボックス」と呼びます。

データボックスは、システム中に複数作成されます。データの操作(格納、抽出など)を行うことができます。

図1.30 データ管理のイメージ



### データボックスの種類

データボックスには以下の種類があります。

- データボックス

- 逐次データの格納に利用する場合

利用者が、データ収集およびデータ配付単位に指定します。業務アプリケーションやシステム間でのキュー制御といった用途として利用できます。

プロセス定義の基本処理、データ収集定義、データ振分定義、およびデータ配付定義で指定できます。

データボックスへの操作方法は、コマンドを利用したデータ格納、抽出、削除などのデータ操作と運用管理クライアントを利用したデータ参照があります。

これらの詳細は、“[コマンドリファレンス](#)”および“[運用管理クライアントヘルプ](#)”を参照してください。

- スプールとして利用する場合

利用者が、データ収集に指定します。データ収集でマスタデータを溜め込んで管理する場合に利用できます。

コマンドを利用したデータ格納、抽出、削除などのデータ操作によって、データボックスをスプールとして扱うことができます。



## ポイント

### データボックスに格納されたデータの削除契機

データボックスに格納されたデータを削除する契機は、以下の2パターンがあります。

- 動作環境ファイルの「自側データボックス蓄積情報の保存期限(excttermキー)」の経過を契機に削除  
動作環境ファイルの「日付を変更する区切り時刻(resettimeキー)」をもとに、データボックスの蓄積情報の保存期限が経過したあと、自動で削除されます。なお、Information Integratorサーバが停止していた場合は次回の起動時に削除されます。動作環境ファイルの詳細は、“セットアップガイド”を参照してください。
- 利用者の削除依頼によって削除  
ifirdelt(データ削除)コマンドの実行によって削除できます。詳細は、“コマンドリファレンス”を参照してください。

### • テンポラリデータボックス

処理プロセスでデータを処理する場合に、データ変換や振分などのファンクションの処理結果データを一時的に保存してファンクション間でのデータの受渡しを行うのに利用します。

ファンクションで異常が発生した場合のリカバリ操作を行ったり、データの流を確認したりすることができます。

一度、抽出したデータを再度、抽出する事はできませんが、データの参照は可能です。

## ポイント

### テンポラリデータボックスに格納されたデータの削除契機

テンポラリデータボックスに格納されたデータを削除する契機は、以下の2パターンがあります。

- プロセスの完了を契機に削除  
動作環境ファイルの「テンポラリデータボックスの削除契機(tmpboxdelキー)」に「proc」を指定した場合および「削除条件(tmpboxdeltermキー)」を指定した場合にプロセスの完了時に自動で削除されます。
- プロセス完了から1日経過後に削除  
動作環境ファイルの「テンポラリデータボックスの削除契機(tmpboxdelキー)」を省略した場合または「fix」を指定した場合、プロセスの完了から1日が経過したあと、動作環境ファイルの「区切り時刻(resettimeキー)」をもとに自動で削除されます。なお、Information Integratorサーバが停止していた場合は次回の起動時に削除されます。

動作環境ファイルの詳細は、“セットアップガイド”を参照してください。

## 管理

「データボックス」には、データが世代として順次蓄積されます。各世代のデータに対して、システムで一意的な番号であるデータ番号が自動採番されます。利用者は、この番号を基にしてデータの識別を行うことができます。

各世代のデータに対して、データの管理情報とデータを以下の構造で管理します。

### • 管理情報

以下の情報を使用して管理します。

- データを示す情報(データ番号、データサイズ、格納日時など)
- データ操作を行った処理(ファンクション)情報
- データの状態

### • データ

データは、動作環境ファイルの「世代管理データ保存領域(stk\_pathキー)」の配下でデータ単位にファイルで管理されます。

## 扱えるデータ

CSVやバイナリといったデータの形式にかかわらず、データボックスに管理することができます。

また、データボックス1つあたりの世代数の上限および各データのサイズにも制限はありません。

## 注意

1つのデータボックスに対して、異なる構造や形式のデータは格納できません。

ただし、以下の場合には異なる構造や形式のデータを混在することができます。

- ・ データボックスに対応する、データ構造定義のデータ形式に「混在フォーマット」を指定した場合

## データボックスの格納状態

データボックスの格納状態は以下のとおりです。これらの各状態は、下記の表の上から順に遷移します。

格納処理、抽出処理で異常を検出した場合は、実行前の状態に戻ります。ファンクションで処理するデータは該当のファンクションが正常終了または異常終了するまでは、格納中または抽出中になります。

表1.12 格納状態

状態	説明
格納中	格納処理中の状態です。データ収集中のデータもこの状態になります。
格納済	格納処理が完了した状態です。抽出することが可能です。
抽出中	抽出処理中の状態です。データ配付中のデータもこの状態になります。
抽出済	抽出処理が完了した状態です。

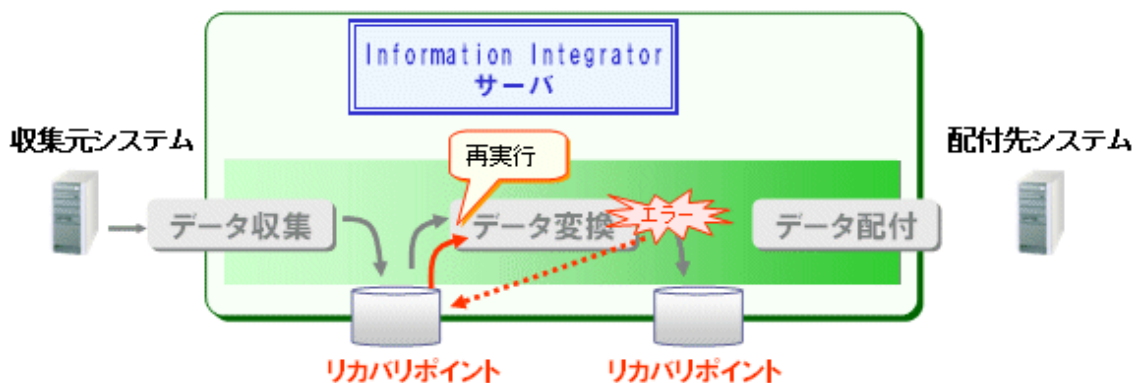
## リカバリポイント

処理プロセスの実行中に、異常検出した場合、処理プロセスを再実行することで、業務をリカバリする必要がありますが、再実行を処理の途中で開始できるように、チェックポイントがあります。このチェックポイントを「リカバリポイント」と呼びます。

リカバリポイントとは、データ収集完了時の処理結果状態およびデータ配付前の処理結果状態を保持した状態を指します。

再実行を依頼された場合には、異常が発生した処理の直前のリカバリポイント(データ収集完了時点または、データ配付前の時点)までの処理をスキップして、処理することで、効率的なリカバリができます。

図1.31 リカバリポイントのイメージ



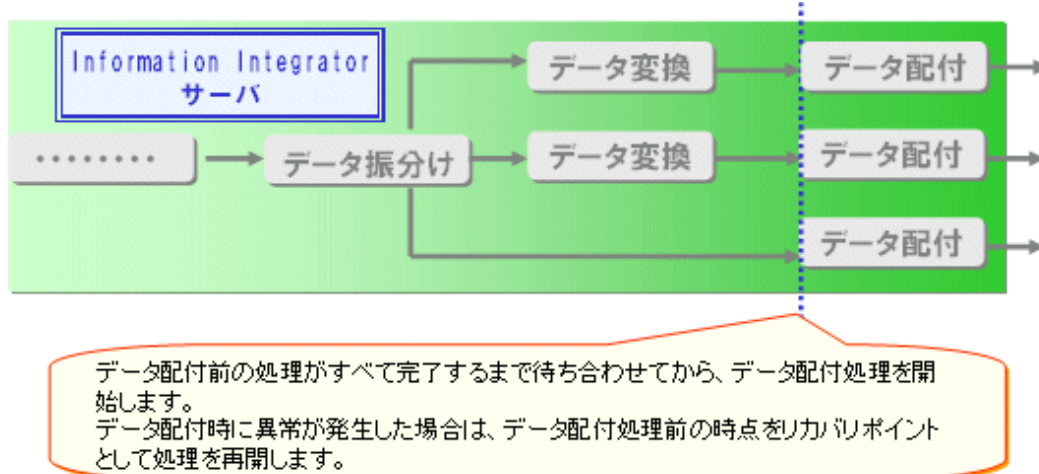
## ポイント

複数のデータ配付処理が存在する処理プロセスのリカバリについて

以下のように、複数のデータ配付処理が存在する処理プロセスでは、データ配付前の処理がすべて完了するまで各データ配付処理の開始を待ち合わせます。

これにより、いずれかのデータ配付処理で異常が発生した場合、異常が発生した処理の直前のリカバリポイント(データ配付前の時点)からリカバリを行うことができます。

図1.32 配付処理の待ち合わせ



### 1.8.3 運用管理

実行環境に登録されている処理プロセスの運用状況を管理します。処理プロセスの運用状況を管理するためのツールを「運用管理クライアント」と呼びます。

運用管理クライアントでは、処理プロセスの予実状況を一覧表示します。また、異常発生時には処理プロセスの処理結果や結果データが確認できるため、異常時の対処が容易にできます。

図1.33 運用管理のイメージ

**運用管理クライアント**

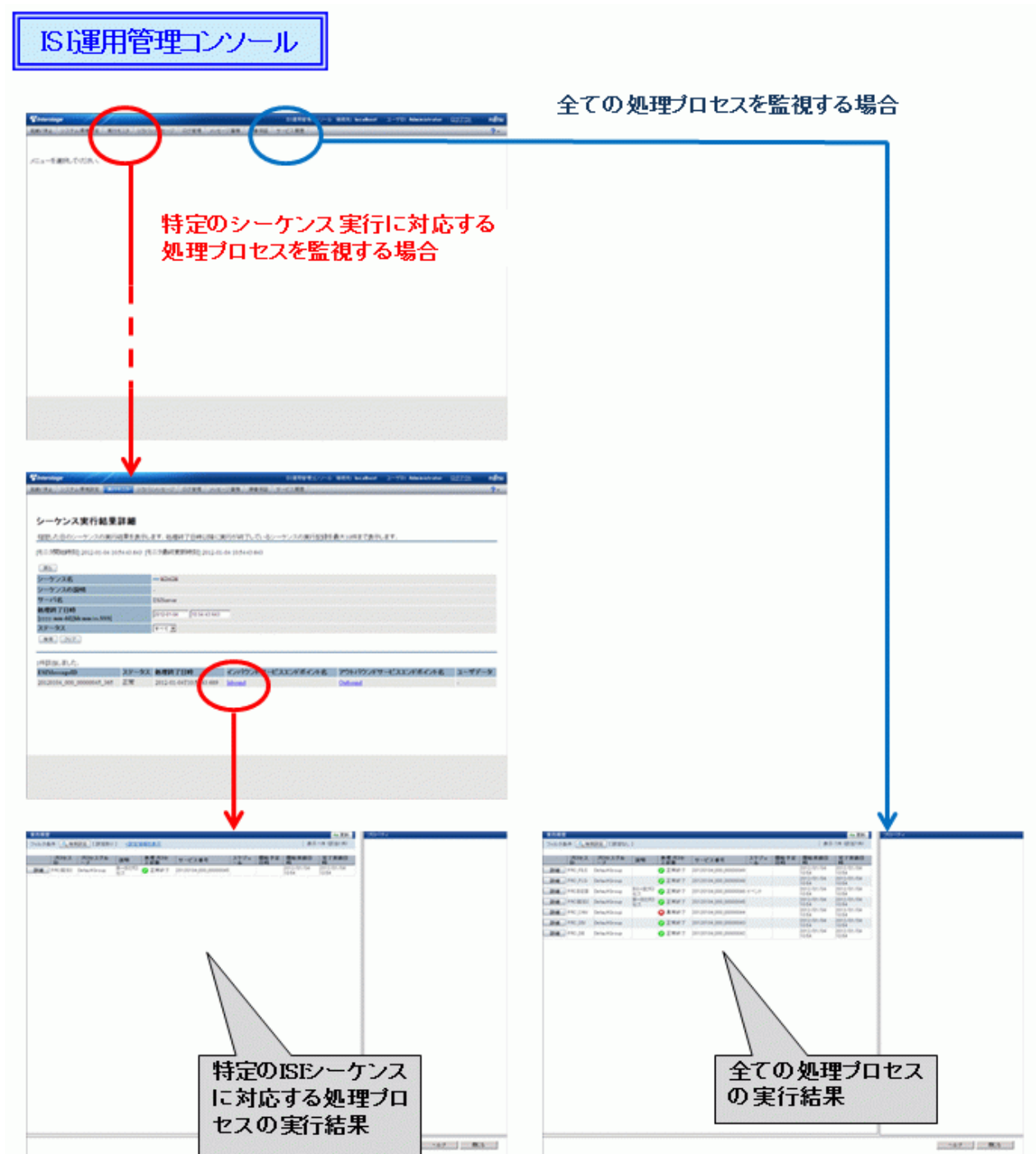
業務の実行予定と実行結果を一覧表示

処理結果の詳細表示

プロセス名	プロセスグループ	説明	実行状況	サービス番号	スケジューム	開始日時	開始実績日時	完了実績日時
PRC_FILE	DefaultGroup		完了	20120104,000,000000005		2012/01/04 12:46	2012/01/04 12:46	2012/01/04 12:46
PRC_DMV	DefaultGroup		異常	20120104,000,000000004		2012/01/04 12:46	2012/01/04 12:46	2012/01/04 12:46
PRC_PLG	DefaultGroup		完了	20120104,000,000000003		2012/01/04 12:46	2012/01/04 12:46	2012/01/04 12:46
PRC_DB	DefaultGroup		完了	20120104,000,000000002		2012/01/04 12:46	2012/01/04 12:46	2012/01/04 12:46
PRC_DV	DefaultGroup		完了	20120104,000,000000001		2012/01/04 12:46	2012/01/04 12:46	2012/01/04 12:46

また、Service Integratorとのサービスバス連携を実施した場合、運用管理クライアントはService IntegratorのISI運用管理コンソールと連動して、特定のISIシーケンス実行に対応する処理プロセス、あるいは、すべての処理プロセスの運用状況を確認できます。(運用管理環境の統合を行った場合)

図1.34 ISI運用管理コンソールと連動する場合の運用管理のイメージ



## 1.9 ファイル転送エージェント機能

ファイル転送エージェント機能は、収集元または配付先のシステム上にあるデータの変換を、その収集元または配付先のシステム上で行うことができる機能です。収集元システムで変換した後にInformation Integratorサーバにファイル形式でデータを配付したり、Information Integratorサーバからファイル形式でデータを配付した後に配付先システムで変換します。Information Integratorサーバの変換機能の一部を切り出すことにより、Information Integratorサーバの負荷分散が実現できます。

また、システム構築、運用、監視までInformation Integratorサーバで一元管理が可能であるため、運用操作性を保持できます。

ファイル転送エージェントと、Information Integratorサーバの機能差については、“ファイル転送エージェントで利用可能な機能一覧”を参照してください。

システム構築、運用、監視の一連の機能、操作については、Information Integratorサーバの各機能説明を参照してください。

## 1.9.1 Information Integratorサーバで一元管理する

---

システム構築、運用、監視までInformation Integratorサーバで一元管理するための特長は、以下のとおりです。

### 定義作成/管理方法

#### 【Information Integratorサーバに統合管理オプションを導入している場合】

処理プロセスの設計情報は、Information Integratorサーバ上で定義の作成および管理を行います。ファイル転送エージェントで利用する定義は、Information Integratorサーバで作成し、ファイル転送エージェントに配付して利用します。

Information Integratorサーバからファイル転送エージェントへ、処理プロセスの定義を配付する方法についての詳細は、“セットアップガイド”を参照してください。

#### 【Information Integratorサーバに統合管理オプションを導入していない場合】

Information Integratorサーバで一元管理することはできません。ファイル転送エージェントにて作成、管理します。作成、管理方法はInformation Integratorサーバと同様です。

### 運用管理

ファイル転送エージェントの処理プロセスの状態はファイル転送エージェントの運用管理クライアントで参照することができます。Information Integratorサーバに統合管理オプションを導入している場合、Information Integratorサーバの運用管理クライアントからファイル転送エージェントの処理プロセスの結果を参照することができます。

運用管理クライアントの詳細については、“運用ガイド”を参照してください。



注意

処理プラグイン機能を利用して、各ファイル転送エージェントの処理プロセスと連携した場合のみ、Information Integratorサーバの運用管理クライアントからファイル転送エージェントの処理プロセスの結果を参照することができます。

## 1.9.2 処理プロセスの運用方法

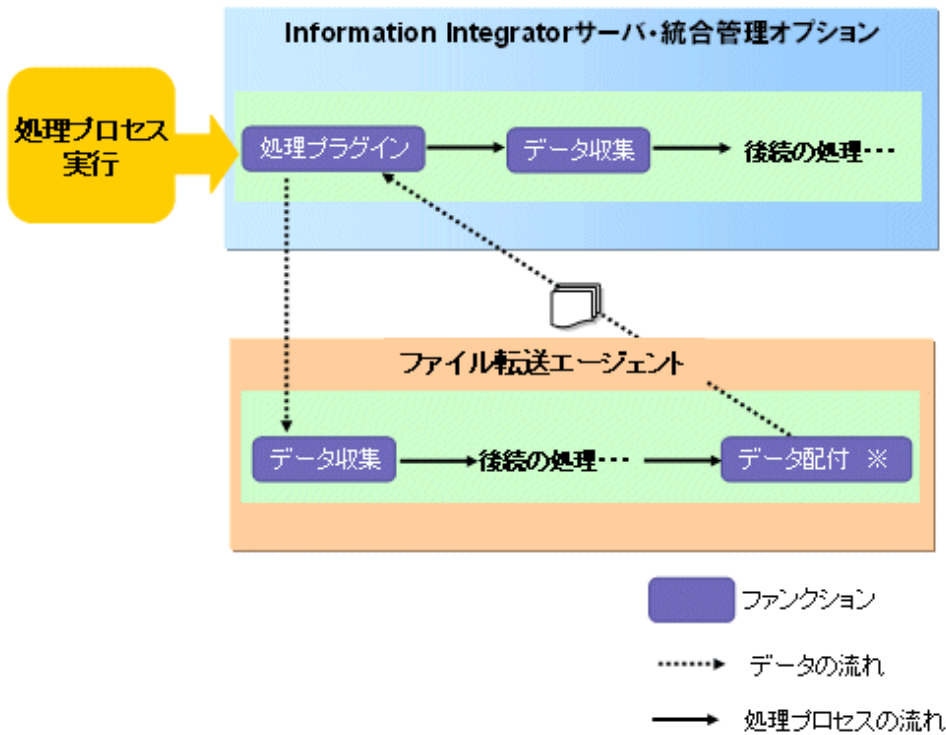
---

ファイル転送エージェントにある処理プロセス、Information Integratorサーバにある処理プロセスは、利用形態に応じて、いずれかのサーバにある処理プロセスを実行することにより、連携できます。

以下に例を示します。



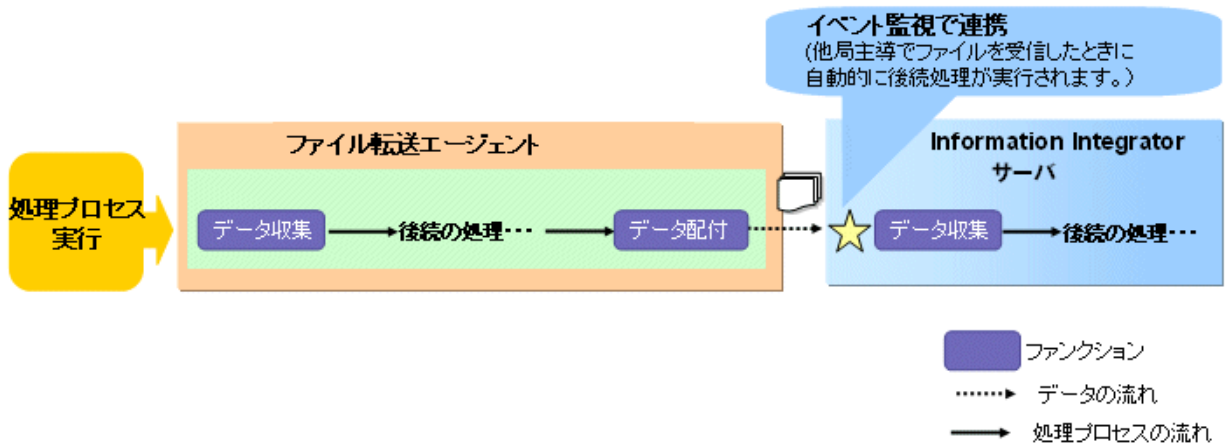
図1.35 例1: Information Integratorサーバの処理プロセスを実行する場合



※:ファイル転送エージェントのデータ配信では、必ずデータはデータボックスに格納します。

Information Integratorサーバの処理プロセスを実行する場合、処理プラグイン機能を利用して、各サーバの処理プロセスを連携します。処理プラグイン機能についての詳細は、III Studioで設計する場合は“6.10.1 処理プラグイン機能の設計手順”、デザインシートで設計する場合は、“7.8.1 処理プラグイン機能の設計手順”を参照してください。

図1.36 例2: ファイル転送エージェントの処理プロセスを実行する場合



ファイル転送エージェントの処理プロセスを実行する場合、Information Integratorサーバの処理プロセスの実行の契機をイベント監視で動作するようにします。これにより、ファイル転送エージェントから配信されたファイルがInformation Integratorサーバに到着した契機で、Information Integratorサーバ側の処理プロセスの実行が開始されるようになります。

ここで利用するイベント監視は、他局主導でファイルを受信したときに自動的に後続処理が実行される機能を利用します。本機能についての詳細は、“1.6.5 イベント監視”を参照してください。

## ポイント

ファイル転送エージェントとInformation Integratorサーバ間のファイル転送形態

ファイル転送エージェントとInformation Integratorサーバ間のファイル転送では、「FTP+」プロトコルを利用するように設計してください。このとき、内部情報の通信を行うため、ファイル送受信の多重度設計は、内部情報の通信を考慮した設計を行う必要があります。詳細は、「8.1.5 ファイル送受信多重度の設計について」を参照してください。

## 1.10 SSL通信を利用する

ここでは、Information IntegratorサーバでSSL通信を行うための環境設定と、証明書管理の運用方法について説明します。

### 1.10.1 証明書管理について

SSL通信では、コネクション接続時に証明書と呼ばれる情報で相互の認証を行います。証明書とは公開鍵が間違いなく利用者本人のものであることを証明するためのもので、証明書を発行した発行局の電子署名が付加されています。

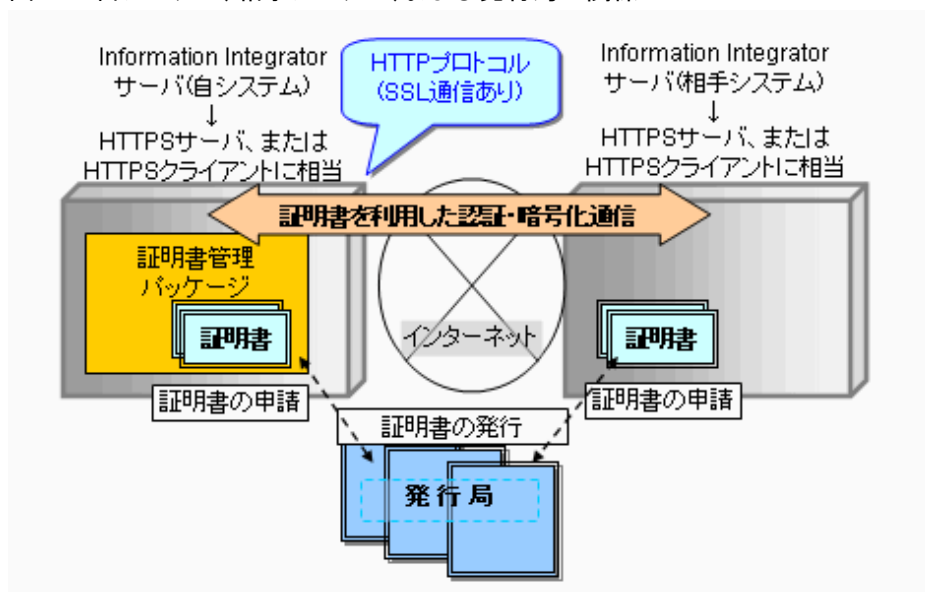
SSLプロトコルバージョンにより認証方法が異なります。

- SSLプロトコルバージョンが2.0の場合  
起動側(クライアント)システムが応答側(サーバ)システムを認証します。
- SSLプロトコルバージョンが3.0の場合  
相互の認証を行います。

相手システムの証明書を証明書に付加された電子署名を含めて検証するには、相手システムの証明書を発行した発行局の証明書が必要です。

自システム、相手システム、および発行局の関係は以下のとおりです。

図1.37 自システム、相手システム、および発行局の関係



- 発行局とは、SSL通信で使用する証明書を発行するシステムのことで、発行局は、「認証局」あるいは「CA局」とも呼ばれます。

#### **P** ポイント

Information Integratorサーバは、以下の製品が発行した証明書のみをサポートしています。

- Systemwalker PKI Manager

- 発行局から取得した証明書は証明書管理パッケージに登録、管理します。SSL通信の際に証明書管理パッケージに管理された情報を元に相手認証と暗号化通信を行います。

## ポイント

証明書管理パッケージ名は、以下のとおりです。

- UNIXの場合: FJSVsmeeパッケージ
- Windowsの場合: F3FSSMEEパッケージ

## 1.10.2 SSL通信のサポート範囲

### SSL通信を利用可能な機能

Information Integratorサーバでは、以下の機能からHTTPプロトコルを利用するときにSSL通信が利用できます。なお、Information Integratorサーバのデータソース定義ではSSL通信を利用したHTTPプロトコルをHTTPSプロトコルとして定義します。

- III studio
- デザインシート

なお、SSL2.0で起動側システムとして通信するInformation Integratorサーバの場合、自システムの証明書の取得は不要です。

### SSL通信のプロトコルバージョンレベル

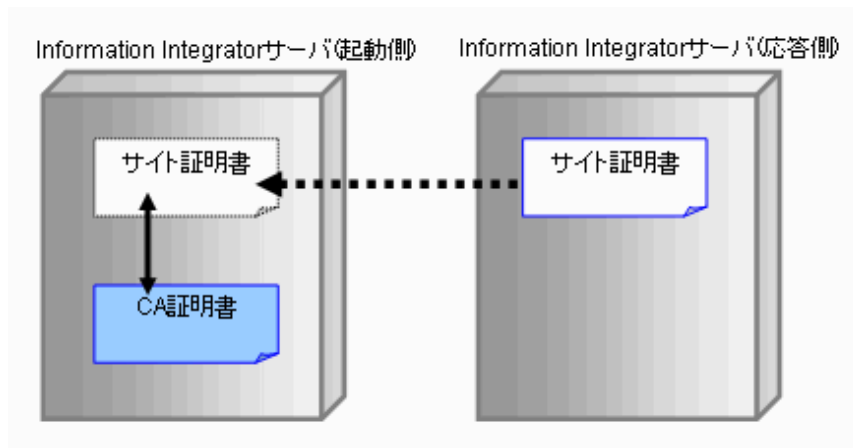
SSL通信にはプロトコルバージョンレベルとしてSSL2.0とSSL3.0があります。

各バージョンレベルの認証プロセスは以下のとおりです。

- SSL2.0

応答側システムの認証を行います。応答側システムは自システムのサイト証明書を取得する必要があります。起動側システムは証明書の取得は必要ありません。ただし、起動側システムは応答側システムの証明書を認証するために応答側システムに対して証明書を発行した発行局の証明書を入手する必要があります。

図1.38 SSL2.0の場合の認証プロセス



## 注意

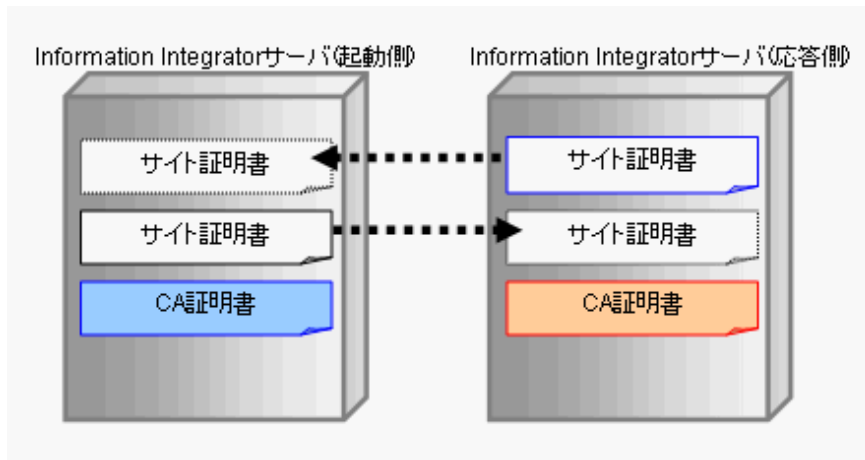
SSL2.0の利用はセキュリティ上推奨されていません。SSL3.0の利用を推奨します。



- SSL3.0

起動側システムと応答側システム双方の認証を行います。どちらのシステムも証明書を取得する必要があります。また、相互に相手システムの証明書を認証するため、相手システムに対して証明書を発行した発行局の証明書を入手する必要があります。

図1.39 SSL3.0の場合の認証プロセス



### SSL通信の相手システムと暗号化通信

Information IntegratorサーバによるSSL通信は相手システムがInformation Integratorサーバ、またはLinkexpressの時に実行可能です。なお、Information Integratorサーバの場合、暗号化だけを行うSSL通信はサポートしていません。SSL通信で転送データの暗号化を行う場合、証明書による相手認証が必要です。

証明書およびSSL通信で利用する暗号化鍵は証明書管理パッケージを利用して管理します。証明書管理パッケージの環境設定については“セットアップガイド”を参照してください。

なお、SSL通信で使用する暗号化鍵はプロトコルバージョンレベルにかかわらず、起動側システムと応答側システムの両方で作成する必要があります。

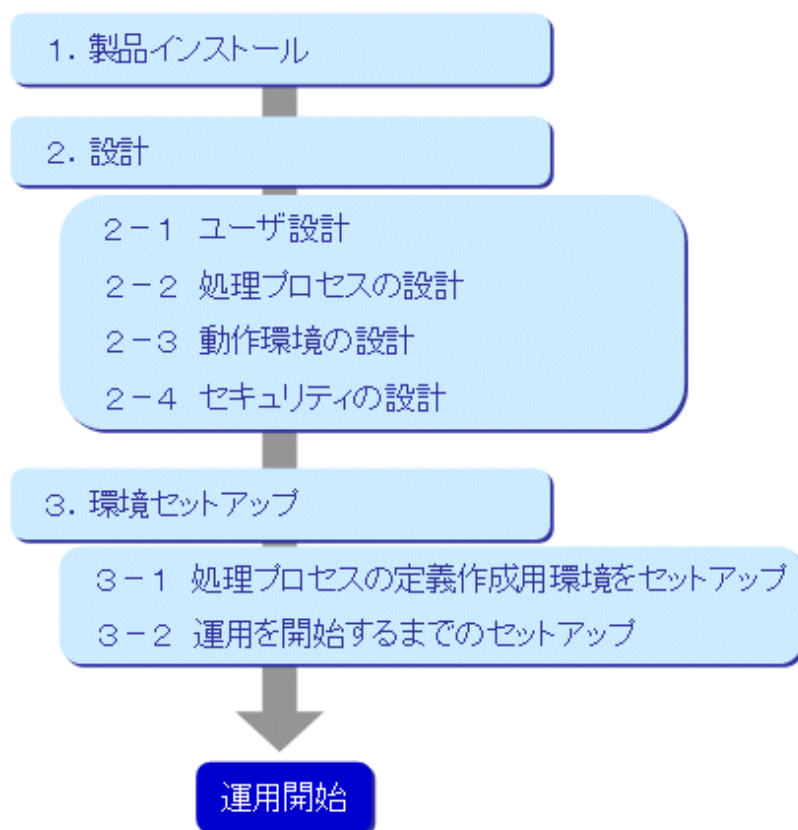
## 第2章 Information Integratorの導入から運用までの流れ

ここでは、Information Integratorの導入から運用までの利用イメージや、利用の流れを説明します。

### 2.1 運用までの流れ

製品インストールから運用までの流れを以下に示します。

図2.1 運用までの流れ



#### 製品インストール

Information Integratorをインストールします。以下をそれぞれインストールしてください。インストール方法については、“ソフトウェア説明書”、または“インストールガイド”を参照してください。

- Information Integratorサーバ
- Information Integratorクライアント

処理プロセスの手順や処理プロセスの実行に必要なパラメタに関する定義の作成および定義の管理を行うツールです。

#### 設計

運用環境を設計します。Information Integratorでは、以下の項目を設計します。

データの収集、変換、配付などデータ統合の一連の処理(処理プロセス)と、環境を設計します。

- ユーザー設計

Information Integratorを利用するユーザーを設計します。

Information Integrator特権ユーザーと、運用ユーザーを用意します。

- 処理プロセスの設計

データの収集、変換、配付などデータ統合の一連の処理(処理プロセス)を設計します。III Studioまたは、デザインシートで行います。

- 動作環境の設計

各処理の多重度制御や、利用するデータベースの環境を設計します。

- セキュリティの設計

システムを運用するためのセキュリティについての設計をします。

## 環境セットアップ

設計に基づいて、Information Integratorの利用環境をセットアップします。

セットアップには、処理プロセスの内容が記述された定義をInformation Integratorサーバに定義登録するまでのセットアップと、Information Integratorサーバの運用を開始するまでのセットアップの2段階があります。

## 運用開始

定義した運用スケジュールに沿って、データ統合処理の運用を開始します。

運用開始後は、運用管理クライアントで、運用が正常に経過していることを確認できます。

## 第3章 ユーザー設計

ここでは、Information Integratorを利用するユーザーの種類について説明します。

### 3.1 ユーザーの種類

Information Integratorで管理するリソース(定義、データ)に対しては、Information Integrator内で利用者権限を設定し、リソース操作ごとに認証することで、リソースの保護を行います。

Information Integratorを利用するためのユーザーの種類には、以下があります。

- Information Integrator特権ユーザー
- Information Integrator運用ユーザー

各ユーザーについて、説明します。

#### Information Integrator特権ユーザー

Information Integratorのすべてのリソース操作が可能な唯一のユーザーです。以降、特権ユーザーと呼びます。特権ユーザーは、以下の条件を満たしている必要があります。



注意

ユーザーIDに以下の文字は利用できません。

半角空白、!、\$、%、&、'、(、)、^、~、{、}、#

#### Windowsの場合

- Administratorsグループに属していること
- 以下のOS権限があること
  - オペレーティングシステムの一部として機能
  - クォータの増加(プロセスのメモリクォータの増加)  
※Administratorsグループであれば、デフォルトで付与されます。
  - サービスとしてログオン
  - トークンオブジェクトの作成
  - プロセスレベルトークンの置き換え

権限を付与する手順を以下に示します。

1. コントロールパネルの管理ツールのローカルセキュリティポリシーを開きます。
2. [セキュリティの設定]-[ローカルポリシー]-[ユーザー権利の割り当て]で該当のポリシーを選択し、利用者IDを追加します。

#### UNIXの場合

rootユーザーを指定してください。

#### Information Integrator運用ユーザー

Information Integratorで提供するコマンドやIII Studio、運用管理クライアント、デザインシートの登録などを操作して運用する、サーバOSにログイン可能な一般ユーザーです。

運用ユーザーに付与される権限(どのリソースに対して、どのような操作が可能か)は、あらかじめ特権ユーザーが定めた設定に従います。

## 注意

ユーザーIDに以下の文字は利用できません。

半角空白、!, \$, %, &, ', (, ), ^, ~, {, }, #

## UNIXの場合

インストール時に指定したグループに属しているユーザーである必要があります。

クラスタ運用時は、必ず運用側、待機側システムともに同じグループを指定してインストールしてください。グループ名(例えば「sys」や「iii」など)ではなく、グループID(数値)を同一にしてください。

- Solarisの場合

「/usr/bin/id -a」の出力結果の「gid(一次グループ)」と「groups(二次グループ)」が同じであることを確認してください。

- Linuxの場合

「/usr/bin/id -a」の出力結果の「gid(一次グループ)」と「所属グループ(二次グループ)」が同じであることを確認してください。

## 例

### インストール時に指定したグループが「iii」(gid=7002)の場合

Information Integrator運用ユーザーとしてのユーザーの利用可否について、「/usr/bin/id -a」の出力結果別に説明します。

なお、以下の出力例は、実際の出力例とは一部異なります。上記の説明に必要な内容のみ抜粋して説明しています。

- 利用可能

「gid」が同一のため、利用可能です。

```
uid=6001(iiiuser) gid=7002(iii) groups=7002(iii),7003(apgrp)
```

- 利用不可

「gid」が異なるため、利用できません。

```
uid=6002(iiiuser2) gid=7003(apgrp) groups=7002(iii),7003(apgrp)
```

## 3.2 各ユーザーの操作範囲

各ユーザーは、以下の操作を行うことが可能です。

操作	操作内容	特権ユーザー	運用ユーザー 権限付与の可否
セットアップ	Information Integratorの動作環境をセットアップします。	○	× 注1
起動/停止	Information Integratorの環境を起動/停止します。以下を起動/停止できます。 <ul style="list-style-type: none"><li>• Information Integratorサーバ</li><li>• Symfoware Server</li></ul>		
定義操作	III Studioまたは、デザインシートで定義を作成して、実行環境に登録します。		○
	III Studioまたは、デザインシートで定義を参照します。		○

操作	操作内容	特権ユーザー	運用ユーザー 権限付与の可否
処理プロセス操作・運用	処理プロセス実行および監視操作を行います。		○ 注2
データ参照	異常時など必要に応じてデータを参照します。		注3

○: 権限付与可能

×: 権限付与不可

必ず特権ユーザーが行う操作です。

注1: Windowsの場合、Administratorグループに属す他のユーザーIDでも実行可能ですが、特権ユーザーと同じOS権限が必要となります。そのため、特権ユーザーで実行するようにしてください。

注2: 処理プロセスをグルーピングし、グループ単位に処理プロセス操作・運用が可能なユーザーを設定します。

処理プロセスグループの設計は、“3.3 処理プロセスの操作・運用を行う運用ユーザーの設計”を参照してください。

注3: 処理プロセスの実行中に格納したデータは、処理プロセスを実行した運用ユーザーのみが参照できます。

### 権限の設定方法

特権ユーザーの設定および定義操作を行う運用ユーザーの権限の設定は、“セットアップガイド”を参照してください。

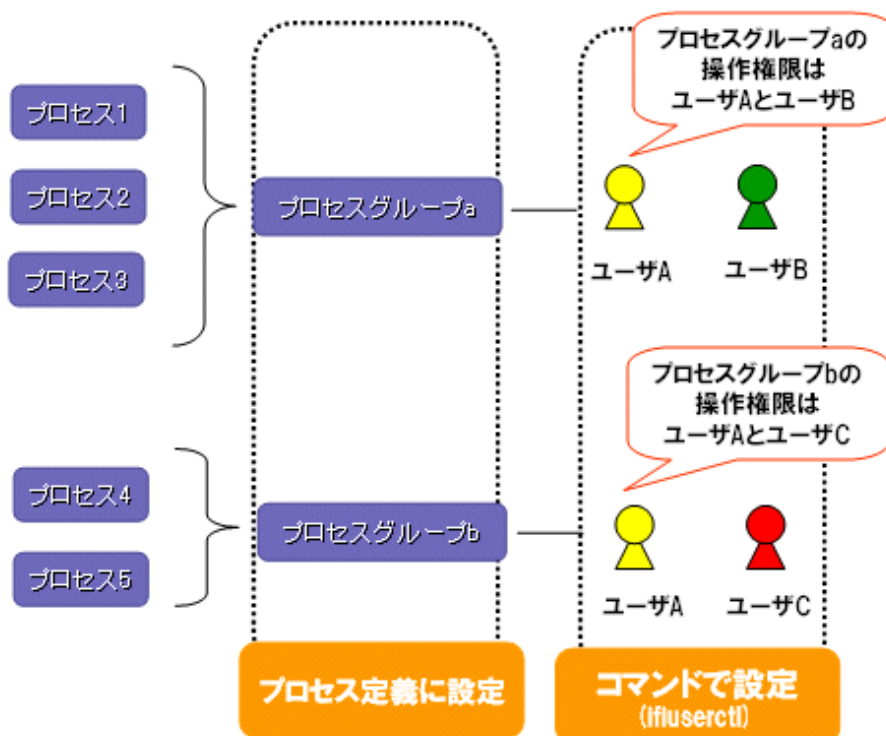
処理プロセス操作・運用を行う運用ユーザーの権限の設定については、“3.3 処理プロセスの操作・運用を行う運用ユーザーの設計”を参照してください。

## 3.3 処理プロセスの操作・運用を行う運用ユーザーの設計

処理プロセスの操作・運用を行う運用ユーザーを決めます。

以下のように、Information Integratorで運用する処理プロセスを、グルーピングします。グルーピングした処理プロセスごとに、操作・運用するユーザーを決めます。

図3.1 運用ユーザーの設計



## 権限の設定方法

処理プロセスのグループIDは、III Studioまたは、デザインシートのプロセス定義に入力します。

また、各プロセスグループに対する運用ユーザーの権限付与は、「ifuserctl(ユーザー権限操作)」コマンドで行います。詳細は、“コマンドリファレンス”を参照してください。

### ポイント

#### 運用管理クライアントで表示される処理プロセス

運用管理クライアントでは、操作・運用が可能な処理プロセスのみ表示されます。上述の例の「ユーザーB」に対しては、運用管理クライアントで「プロセスグループa」に属するプロセス1～プロセス3は表示され、「プロセスグループb」に属するプロセス4およびプロセス5は表示されません。

### 注意

#### 統合管理オプションとファイル転送エージェントを併用する場合

統合管理オプションとファイル転送エージェントを併用する場合、ファイル転送エージェントごとに一意なプロセスグループ名を設定し、各ファイル転送エージェントに必要な定義のみが参照できるようにしてください。

統合管理オプションとファイル転送エージェントを併用する場合についての詳細は、“セットアップガイド”を参照してください。

## 第4章 システム構成

ここでは、Information Integratorを利用するためのシステム構成について説明します。

### 4.1 システム構成

Information Integratorを利用する場合のシステム構成について説明します。

#### 4.1.1 Information Integratorの構成物

Information Integratorを構成するコンポーネントを説明します。

##### Information Integratorサーバ

Information Integratorを利用するためのサーバです。

抽出、変換、配付の一連のプロセスを処理します。

##### 運用管理用Webサーバ

運用管理クライアントを利用するために必要なサーバです。

##### III Studio

視覚的に確認しながら設計を進めることが可能な定義作成ツールです。GUIベースの統合開発環境Interstage Studio上で動作するプラグインです。

定義管理クライアント、デザインシートと合わせて、定義クライアントと呼びます。

また、定義管理クライアント、デザインシート、運用管理クライアントと合わせて、Information Integratorクライアントと呼びます。

##### 定義管理クライアント

デザインシートを作成、登録するためのクライアントです。

III Studio、デザインシートと合わせて、定義クライアントと呼びます。

また、III Studio、デザインシート、運用管理クライアントと合わせて、Information Integratorクライアントと呼びます。

##### デザインシート

Excelの操作で容易に入力が可能な定義作成ツールです。Excel上で動作するプラグインです。

III Studio、定義管理クライアントと合わせて、定義クライアントと呼びます。

また、III Studio、定義管理クライアント、運用管理クライアントと合わせて、Information Integratorクライアントと呼びます。

##### 運用管理クライアント

登録した処理プロセスの状態を管理するためのクライアントです。

III Studio、定義管理クライアント、デザインシートと合わせて、Information Integratorクライアントと呼びます。

##### 相手側システム

- ・ 収集元システム

収集元のデータを保持しているシステムを指します。

Information Integratorでは、POS、会計などの業務システムや基幹システム、各種パッケージ製品など、幅広いシステムのデータを対象にしています。

- ・ 配付先システム

Information Integratorで、収集・変換したデータを転送する先のシステムを指します。



収集・変換したデータは、情報活用、帳票出力や各種パッケージでの利用など様々なシステムで利用できます。

### ファイル転送エージェント(オプション製品)

Information Integratorを利用するためのサーバです。Information Integratorサーバとは別サーバで動作し、抽出、変換、配付の一連のプロセスを処理します。

収集元または配付先のシステム上にあるデータの変換を、その収集元または配付先のシステム上で行うためのサーバです。収集元システムで変換した後にInformation Integratorサーバにデータを配付、または、Information Integratorサーバからデータを収集した後に変換します。

### 統合管理オプション(オプション製品)

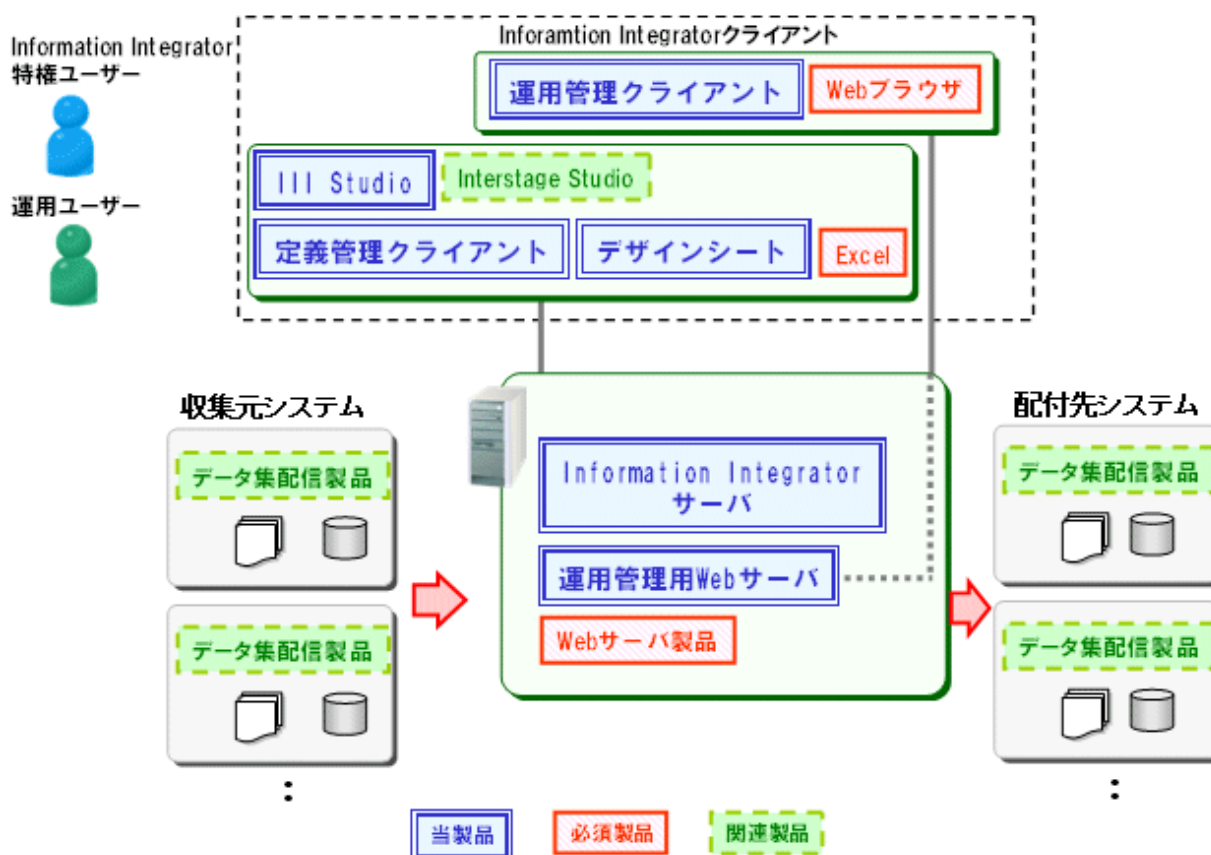
Information Integratorサーバと同一サーバで動作し、Information Integratorとサービスバスが連携する業務の構築・運用を支援したり、ファイル転送エージェントを導入した連携先も含めた定義・運用の一元管理をしたりします。

## 4.1.2 標準構成

Information Integratorを利用する場合の標準的な構成について説明します。標準的な構成では、定義作成ツールであるIII Studioと、デザインシートを必要に応じて併用するパターンを想定しています。

相手側システムのデータ集配信製品については、“4.3 相手側システムの種類ごとのシステム構成”を参照してください。

図4.1 Information Integratorを利用する場合の標準的な構成



- Windowsサーバの場合、Information Integratorサーバ、およびInformation Integratorクライアントは、同一サーバに導入することができます。
- データ集配信製品は、利用側システム、利用機能に応じて、選択してください。接続可能なデータ集配信製品については、“4.2.1.2 Information Integratorが接続可能なデータ集配信製品”を参照してください。

## P ポイント

定義管理、定義作成、定義修正をすべてIII Studioで行う場合は、以下の機能は不要です。

- ・ 定義管理クライアント
- ・ デザインシート
- ・ Excel

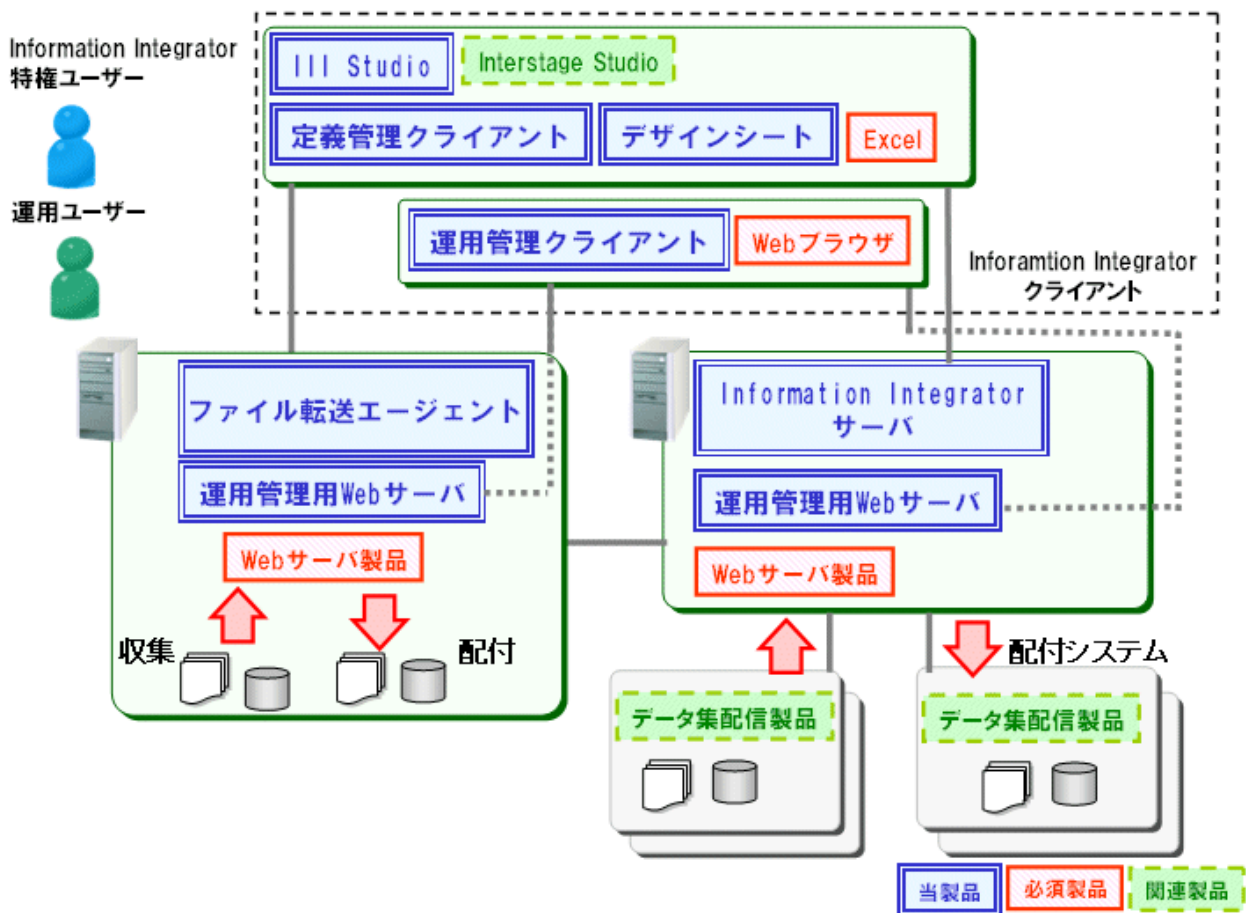
定義作成、定義修正をすべてデザインシートで行う場合は、以下の機能は不要です。

- ・ III Studio (Interstage Studio)

### 4.1.3 負荷分散構成

ファイル転送エージェントを利用して、Information Integratorサーバの負荷分散を実現する構成について説明します。

図4.2 負荷分散構成



- ・ Windowsサーバの場合、Information Integratorサーバ、およびInformation Integratorクライアントは、同一サーバに導入することができます。ファイル転送エージェントとInformation Integratorクライアントも、同一サーバに導入することができます。
- ・ データ集配信製品は、利用側システム、利用機能に応じて、選択してください。接続可能なデータ集配信製品については、“4.2.1.2 Information Integratorが接続可能なデータ集配信製品”を参照してください。
- ・ ファイル転送エージェントで利用可能な機能についての詳細は、“付録C ファイル転送エージェントで利用可能な機能一覧”を参照してください。

## P ポイント

定義管理、定義作成、定義修正をすべてIII Studioで行う場合は、以下の機能は不要です。

- ・ 定義管理クライアント
- ・ デザインシート
- ・ Excel

定義作成、定義修正をすべてデザインシートで行う場合は、以下の機能は不要です。

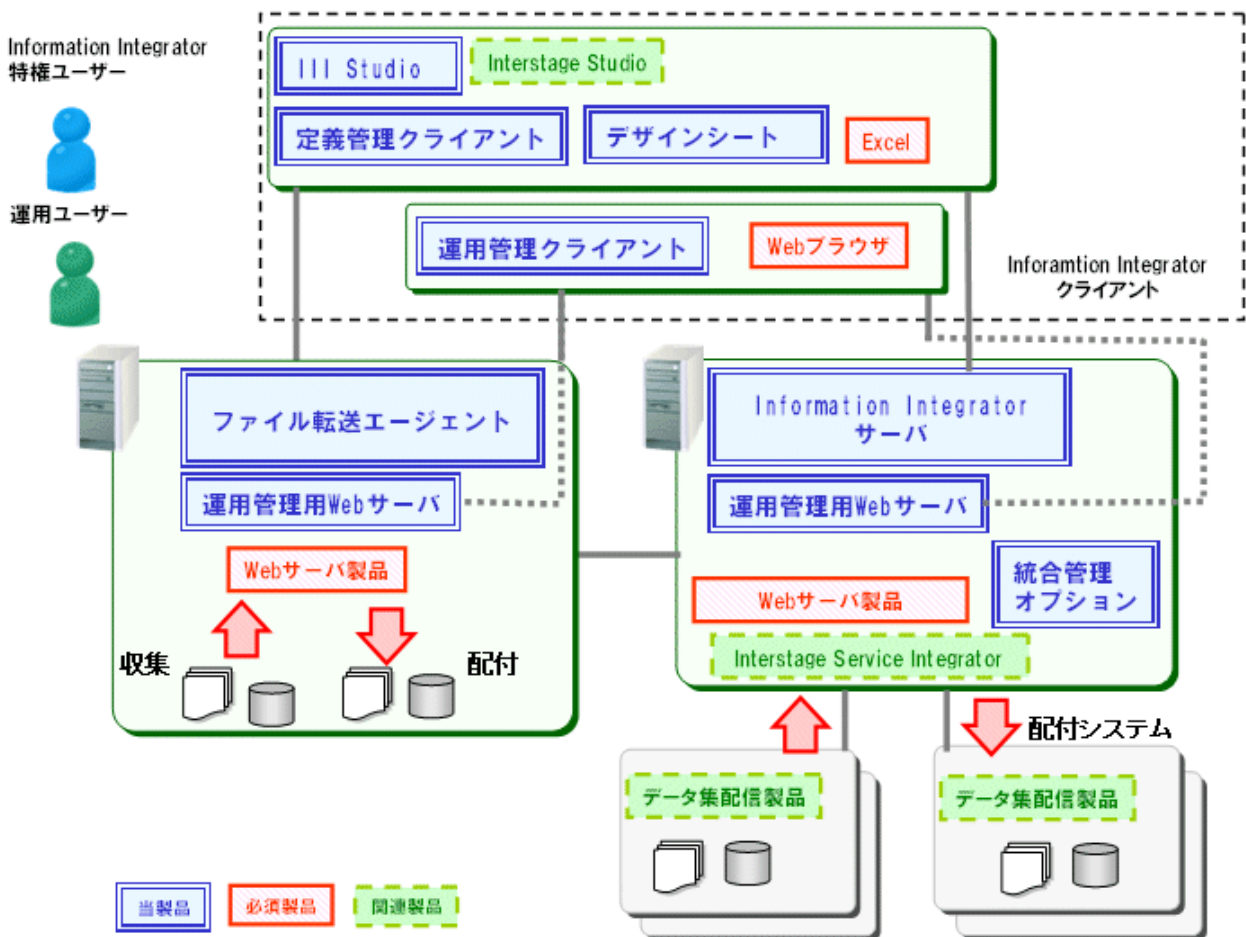
- ・ III Studio (Interstage Studio)

### 4.1.4 Information Integrator統合管理構成

統合管理オプションを利用して、Service Integratorとの連携を行う構成です。

以下は、ファイル転送エージェントを併用する構成例です。

図4.3 Information Integrator統合管理構成

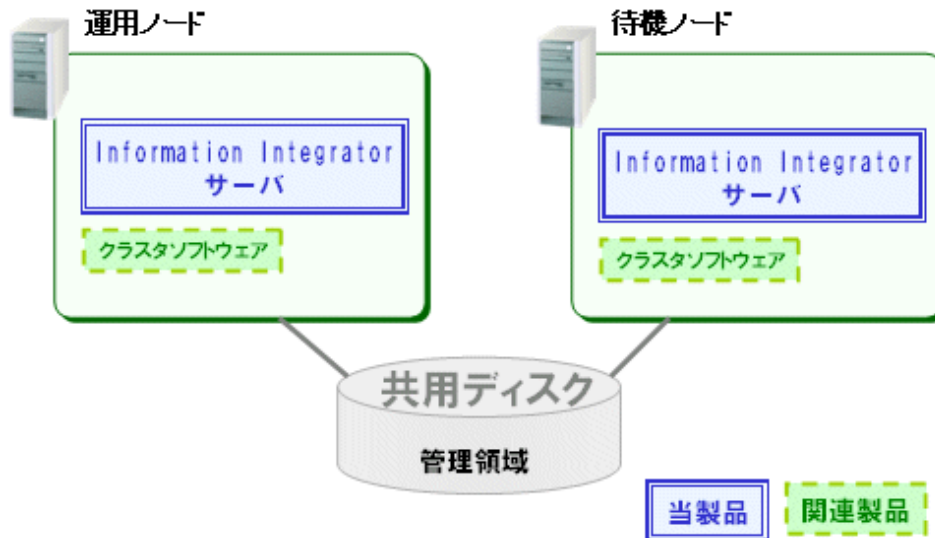


### 4.1.5 クラスタ構成

Information Integratorサーバを1:1運用待機のクラスタ構成にできます。

Information Integratorサーバをクラスタ構成にする場合は、運用ノード、待機ノードに、それぞれInformation Integratorサーバを導入します。

図4.4 Information Integratorのクラスタ構成



Information Integratorクライアント、および相手側システムからの通信は、「論理IPアドレス」を指定します。

管理領域は、共用ディスク上にセットアップします。

管理領域については、“セットアップガイド”を参照してください。

## 4.2 相手側システムの接続について

Information Integratorで利用できる相手側システム(収集元システム、配付先システム)について説明します。

### 4.2.1 ファイル送受信の接続について

ファイル送受信の場合に、Information Integratorで利用できる相手側システム(収集元システム、配付先システム)との接続について説明します。

#### 4.2.1.1 ファイル送受信で扱える通信プロトコル

ファイル送受信(ファイル受信、ファイル送信)で扱える通信プロトコルを以下の表に示します。相手側システムの環境に合わせて、利用するプロトコルを選択してください。

表4.1 ファイル送受信で扱える通信プロトコルの種類

通信プロトコル(※1)		説明
TCP/IP	FTP+	<p>標準FTPにファイルの送信、受信確認などを拡張した富士通独自の通信プロトコルです。</p> <p>以下の特長があります。</p> <ul style="list-style-type: none"> <li>標準FTPでの信頼性問題を解決、高信頼性プロトコル</li> <li>拡張機能をサポート、データ転送機能に付加価値</li> </ul> <p>本プロトコルを利用する場合、相手側システムにLinkexpressが必要です。</p> <p>相手側システムにLinkexpressが導入されている場合は、本プロトコルの利用を推奨します。</p> <p>Information Integratorサーバとファイル転送エージェントの接続は、本プロトコルを利用してください。この場合、相手側システムにLinkexpressの導入は不要です。</p>
	FTP	<p>TCP/IPのファイル通信プロトコルです。</p> <p>相手側システムにLinkexpressが導入されていない場合、利用してください。</p>

通信プロトコル(※1)		説明
	FTPM	相手側システムがASPの場合に利用する富士通独自の通信プロトコルです。 本プロトコルを利用する場合は、相手側システムに、DTSが必要です。
	HTTP(※2)	TCP/IPのハイパーテキスト通信プロトコルです。 システムのセキュリティポリシーにより、FTP+/FTPプロトコルを使用できない場合に本プロトコルを利用します。DMZゾーンとイントラネット間でのファイル転送です。
	HTTPS	SSL通信を実装したHTTPプロトコルです。 HTTPプロトコルについては、上記HTTPの説明を参照してください。
HICS(FNA)		富士通独自の通信プロトコルFNAのファイル通信プロトコルです。 FNA3手順の1次側をサポートしているグローバルサーバ/PRIMEFORCE、およびASPと、FNA接続する場合に利用します。
SAN		ディスクアレイ装置を利用したSAN環境上で使用する富士通独自のファイル通信プロトコルです。 LANを利用しないため、大容量ファイル転送、およびセキュリティ保持に優れます。 本プロトコルを利用する場合、相手側システムに、Linkexpressが必要です。
HULFT		相手側システムがHULFTの場合のファイル通信プロトコルです。

※1:利用するプロトコルによって、使用できる機能が異なります。プロトコルごとの機能範囲については、“[A.1 通信プロトコルごとの利用可能機能一覧](#)”を参照してください。

※2:HTTPプロトコルを利用する場合、以下の注意事項があります。

**Information Integratorが起動側動作の場合**

応答側システムは、Information Integrator以外のHTTPサーバとの接続も可能です。

**Information Integratorが応答側動作の場合**

起動側システムは、Information IntegratorまたはLinkexpressが必須です。WEBクライアントからの接続はできません。

なお、本プロトコルを利用する場合は、接続相手システムにInformation IntegratorまたはLinkexpressの導入を推奨します。



**注意**

Linuxでは、HICS(FNA)、SANは、利用できません。

なお、すべての転送プロトコルは、冗送設計となっています。冗送設計とは、ファイル送受信の起動側システム(ファイル送受信の要求側)と応答側システムで正常/異常の判断が異なる場合、起動側システムで異常を検知します。起動側システムで異常を検知した場合、ファイル送受信のリトライ処理が可能となり、応答側システムでは二重(冗送)でデータを受信することがあります。

このため、冗送が発生することを考慮して、システム運用の設計を行ってください。

ただし、FTPプロトコルを除くその他のプロトコルでは、最終応答電文の通知に失敗した場合のみ冗送が発生します。最終応答電文の通知失敗は非常に稀なタイミングです。

#### 4.2.1.2 Information Integratorが接続可能なデータ集配信製品

表4.2 Information Integratorが接続可能なデータ集配信製品の種類

製品名称	プラットフォーム	プロトコル一覧(※1)								備考
		FTP+	FTP	FTP M	HTTP	HTT P S	HICS (FNA)	S A N	HULFT	
Linkexpress	Windows Linux Solaris	○	○	○	×	×	○	×	×	

製品名称	プラットフォーム	プロトコル一覧(※1)								備考	
		FTP+	FTP	FTP M	HTTP	HTT P S	HICS (FNA)	SA N	HULFT		
Linkexpress Advanced Edition	Windows Linux Solaris	○	○	○	○	○	○	○	×	×	
Linkexpress SAN option	Windows Linux Solaris OS IV/MSP OS IV/XSP	×	×	×	×	×	×	×	○	×	
Linkexpress Standard Edition	Windows Linux	○	×	○	×	×	○	×	×	×	
Linkexpress Enterprise Edition	Windows Linux	○	○	○	○	○	○	○	○	×	
Linkexpress File Transfer	OS IV/MSP OS IV/XSP	○	○	×	×	×	○	×	×	×	
DTS	OS IV/MSP OS IV/XSP	○	○	○	×	×	×	×	×	×	FTPMプロトコルは、BFTPと います。
	ASP	×	×	○	×	×	×	×	×	×	
HICS	OS IV/MSP OS IV/XSP	×	×	×	×	×	○	×	×	×	
	ASP	×	×	×	×	×	○	×	×	×	
HULFT	—	×	×	×	×	×	×	×	×	○	
標準FTPツール ※2	—	×	○	×	×	×	×	×	×	×	

※1: 利用する通信プロトコルによって、使用できる機能が異なります。通信プロトコルごとの機能範囲については、“[A.1 通信プロトコルごとの利用可能機能一覧](#)”を参照してください。

※2: Information IntegratorがサポートするFTPコマンドの範囲で接続する必要があります。Windows上で動作するFTPクライアントツールは、接続できません。Information IntegratorがサポートするFTPコマンドの範囲は、“[付録E Information IntegratorがサポートするFTPのコマンド](#)”を参照してください。

#### 4.2.1.3 ファイルシステムの留意事項について

Information Integratorサーバのインストールディレクトリ、およびInformation Integratorサーバがアクセスする送受信ファイルの格納ディレクトリ(相手側システム)のファイルシステムの留意事項について説明します。

##### Windowsの場合

Windowsの場合について説明します。

- Information Integratorがサポートしているファイルシステムは、NTFSです。FAT32ファイルシステムは、利用できません。
- 圧縮属性のドライブ、フォルダ、およびファイルは利用しないでください。圧縮属性の場合、通常のI/O処理と比較して、性能上のオーバーヘッドが発生します。このため、I/O量に依存してOSリソースの不足によるアクセスエラーが発生することがあります。  
※圧縮属性は、エクスプローラ上で該当のドライブ、フォルダ、およびファイルのプロパティで、確認、変更できます。
- 暗号化属性のフォルダ、およびファイルは利用しないでください。暗号化属性の場合、Information Integratorサーバの実行者とフォルダ、およびファイルの作成者が異なるためアクセスエラーが発生します。  
※暗号化属性は、エクスプローラ上で該当のフォルダ、およびファイルのプロパティでの確認、変更できます。



## Solarisの場合

Solarisの場合について説明します。

- ・ ZFSファイルシステムは、利用できません。

## 4.2.2 オブジェクト抽出・格納時の通信プロトコル

Salesforceのオブジェクト抽出・格納では、SalesforceのWebサービスを介して、SalesforceとHTTPSで通信します。

## 4.2.3 BLOB収集・配付時の通信プロトコル

BLOB収集・配付では、RESTを利用して通信します。

## 4.3 相手側システムの種類ごとのシステム構成

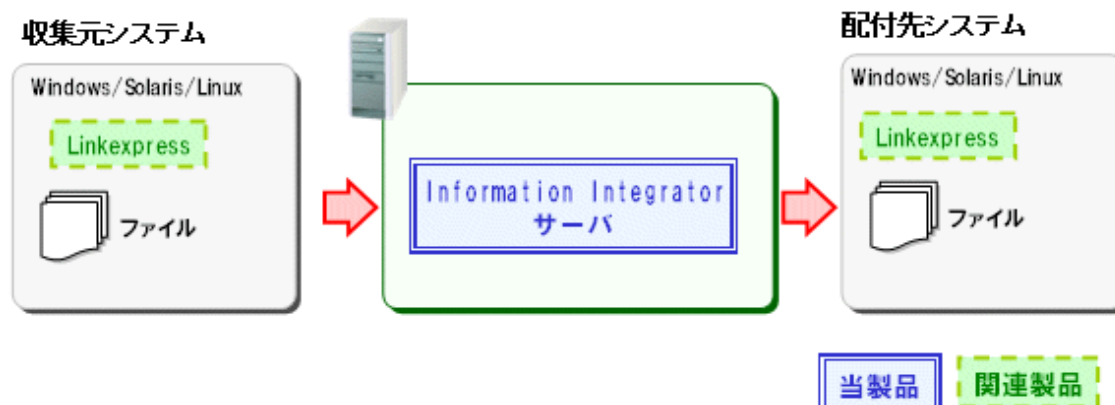
ここでは、相手側システムの種類ごとに、必要なデータ集配信製品、関連製品について説明します。

### 4.3.1 相手側システムがWindows、Solaris、Linuxの場合

相手側システムがWindows、Solaris、またはLinuxの場合、相手側システムに、Linkexpress製品を導入します。

#### ファイル送受信を利用する場合

図4.5 相手側システムがWindows、Solaris、Linuxの場合に、ファイル送受信を利用する場合の構成イメージ



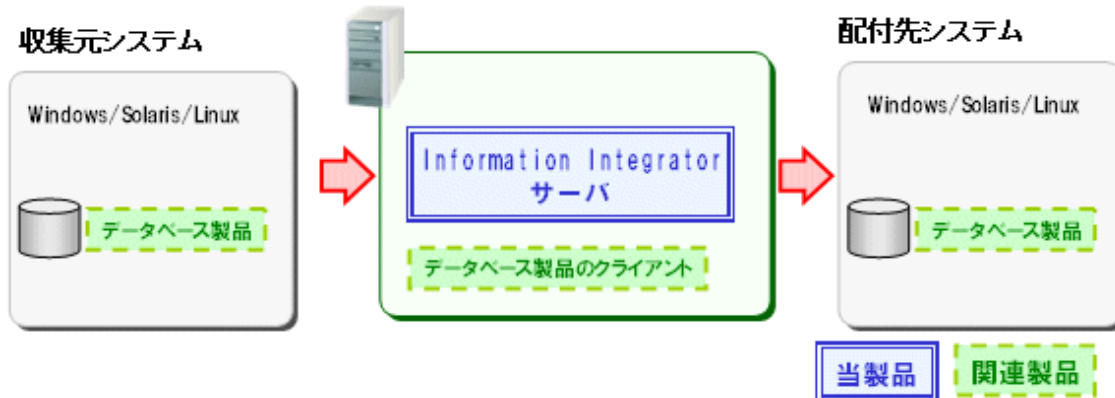
相手側システムが、Windows/Solaris/Linuxの場合に、利用可能なLinkexpressは、以下のとおりです。

- ・ Linkexpress
- ・ Linkexpress Advanced Edition
- ・ Linkexpress Standard Edition
- ・ Linkexpress Enterprise Edition

#### DB抽出、格納を利用する場合

DB抽出、DB格納を行う場合は、Information Integratorサーバ側に、データベース製品のクライアントが必要です。

図4.6 相手側システムがWindows、Solaris、Linuxの場合に、DB抽出・格納を利用する場合の構成イメージ



データベース製品のクライアントには、それぞれ以下のコンポーネントを導入してください。

表4.3 データベース種別とデータベース製品のクライアントの関係

データベース種別	データベース製品のクライアント
Symfoware Server	Symfoware Client
Oracle	Oracle Database Client
SQL Server/SQL Azure	SQL Server Native Client
DB2	IBM Data Server Client
PostgreSQL	postgresql-odbc
その他のデータベース(ODBC接続)	データベースに付属するODBCドライバ

### 注意

Oracle、SQL Server、DB2、PostgreSQL、SQL Azure、その他のデータベースを使用する場合は、データベース製品のクライアントを導入後に、ODBCドライバを使ってDBMSに接続可能な状態としてください。接続方法については、各DBMSのマニュアルを参照してください。

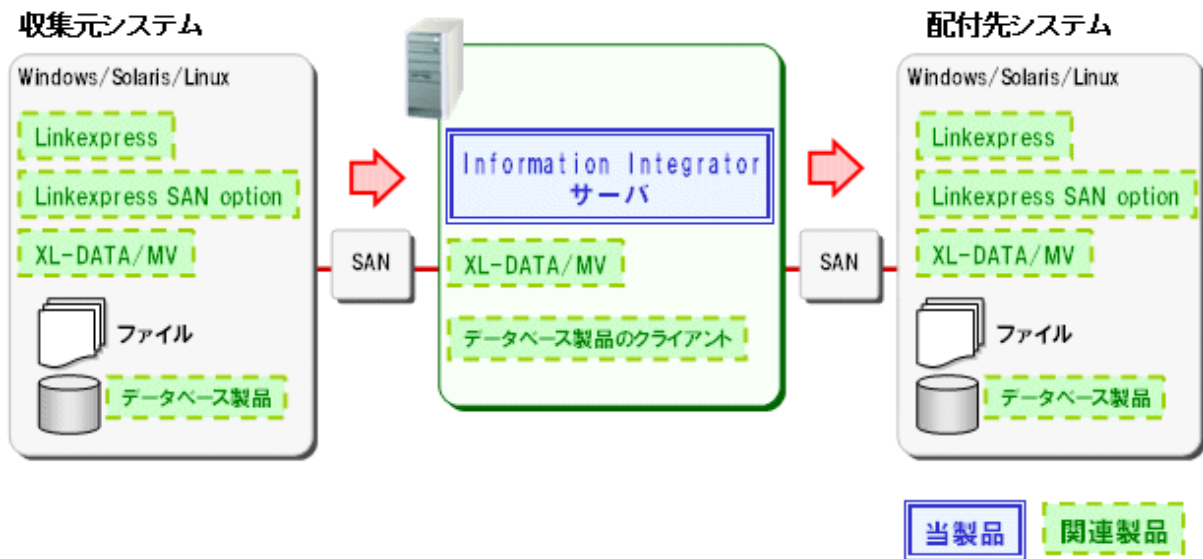
### 通信プロトコルがSANの場合

相手側システムがWindows、Solaris、またはLinuxの場合、相手側システムに、Linkexpress、およびLinkexpress SAN optionを導入します。

また、SANを利用する場合は、利用する相手側システムと、Information Integratorサーバの導入システムの両方に、XL-DATA/MVを導入します。



図4.7 相手側システムがWindows、Solaris、Linuxの場合に、通信プロトコルにSANを利用する場合の構成イメージ



通信プロトコルがSANの場合に利用可能なLinkexpressは、以下のとおりです。

- Linkexpress
- Linkexpress Advanced Edition
- Linkexpress Enterprise Edition

**P** ポイント

**Linkexpress SAN option**について

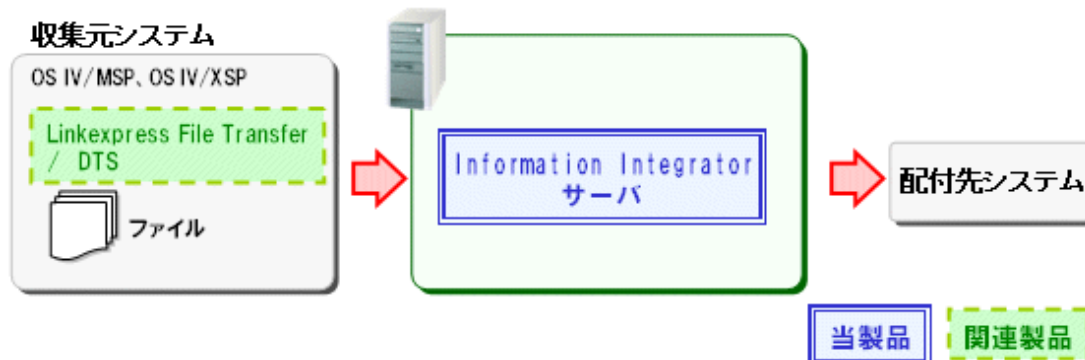
Linkexpress SAN optionは、以下のLinkexpress製品を利用する場合に必要です。

- Linkexpress
- Linkexpress Advanced Edition

**4.3.2 相手側システムがグローバルサーバ/PRIMEFORCEの場合**

相手側システム(収集元システム)がグローバルサーバ/PRIMEFORCE(OS IV/MSP、OS IV/XSP)の場合は、収集元システムに、Linkexpress File Transfer、またはDTSを導入します。

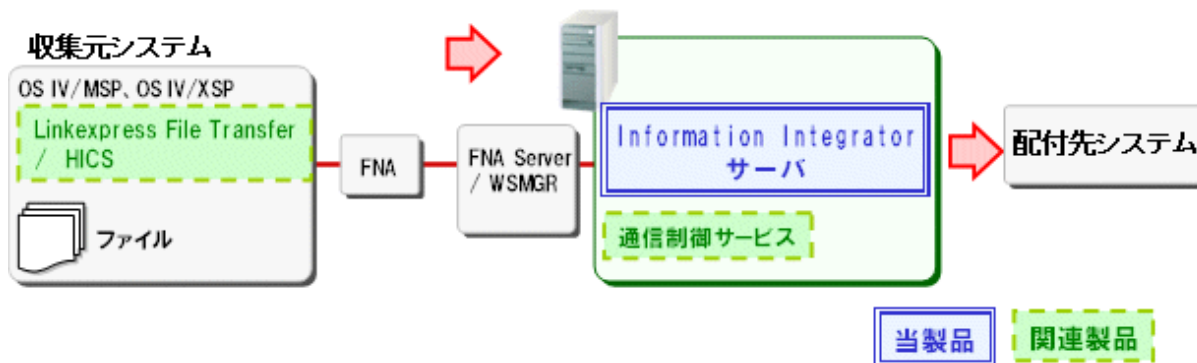
図4.8 相手側システムがグローバルサーバ/PRIMEFORCEの場合の構成イメージ



### 通信プロトコルがFNAの場合

相手側システム(収集元システム)がグローバルサーバ/PRIMEFORCE(OS IV/MSP、OS IV/XSP)の場合で、通信プロトコルにFNAを利用する場合は、収集元システムに、Linkexpress File Transfer、またはHICSを導入し、Information Integratorサーバの導入システムに、通信制御サービスを導入します。

図4.9 相手側システムがグローバルサーバ/PRIMEFORCEの場合に、通信プロトコルにFNAを利用する場合の構成イメージ

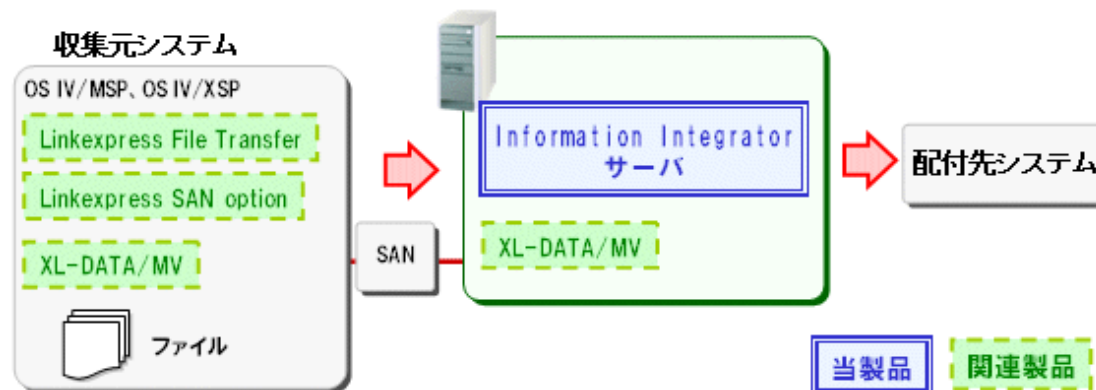


### 通信プロトコルがSANの場合

相手側システム(収集元システム)がグローバルサーバ/PRIMEFORCE(OS IV/MSP、OS IV/XSP)の場合で、通信プロトコルにSANを利用する場合は、収集元システムに、Linkexpress File Transfer、およびLinkexpress SAN optionを導入します。

また、SANを利用する場合は、収集元システムと、Information Integratorサーバの導入システムの両方に、XL-DATA/MVを導入します。

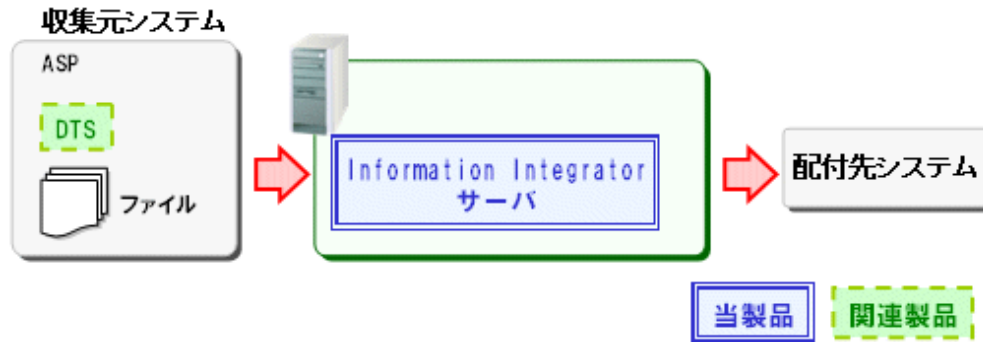
図4.10 相手側システムがグローバルサーバ/PRIMEFORCEの場合に、通信プロトコルにSANを利用する場合の構成イメージ



### 4.3.3 相手側システムがASPの場合

相手側システム(収集元システム)がASPの場合は、収集元システムにDTSを導入します。

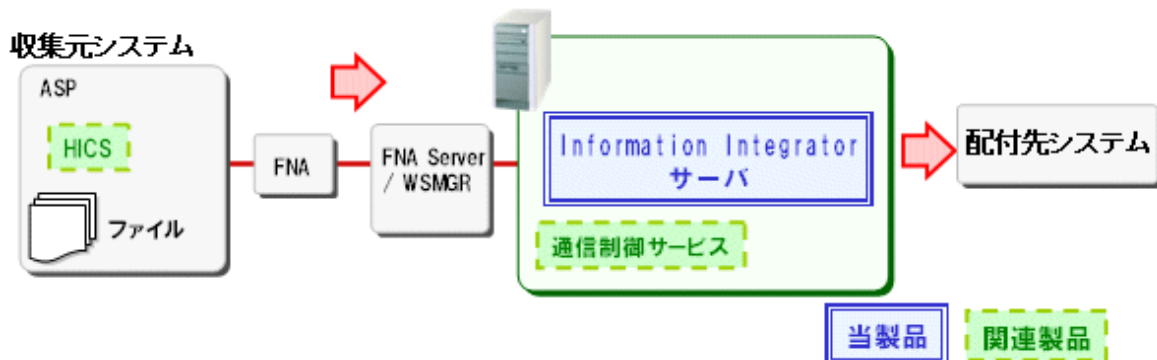
図4.11 相手側システムがASPの場合の構成イメージ



#### 通信プロトコルがFNAの場合

相手側システム(収集元システム)がASPの場合で、通信プロトコルにFNAを利用する場合は、収集元システムにHICSを導入し、Information Integratorサーバの導入システムに、通信制御サービスを導入します。

図4.12 相手側システムがASPの場合に、通信プロトコルにFNAを利用する場合の構成イメージ

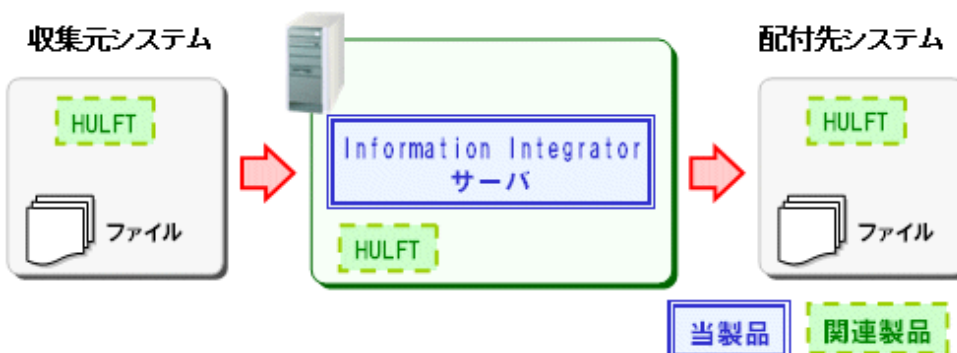


### 4.3.4 HULFTを利用する場合

HULFTを利用する場合は、相手側システムと、Information Integratorサーバの導入システムに、HULFTを導入します。

HULFTが導入できるプラットフォームについては、HULFTのマニュアルを参照してください。

図4.13 HULFTを利用する場合の構成イメージ

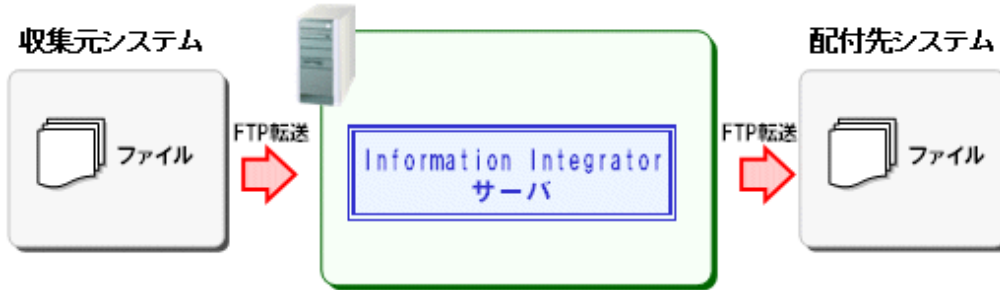


### 4.3.5 標準FTPツールを利用する場合

標準FTPツールを利用する場合は、特にデータ集配信製品は必要ありません。

また、相手側システムのプラットフォームについては、FTPツールが使える環境であれば、特に制限はありません。

図4.14 標準FTPツールを利用する場合の構成イメージ



### 4.3.6 相手側システムがSalesforceの場合

相手側システムがSalesforceの場合は、特にデータ集配信製品は必要ありません。

図4.15 収集元システムがSalesforceの場合の構成イメージ

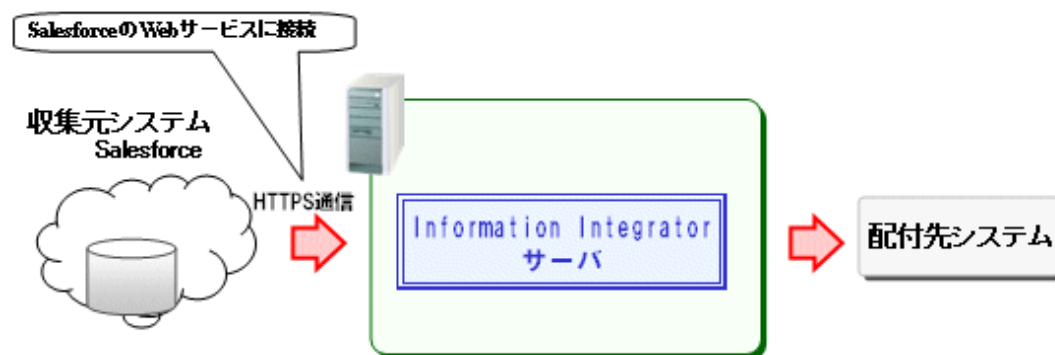
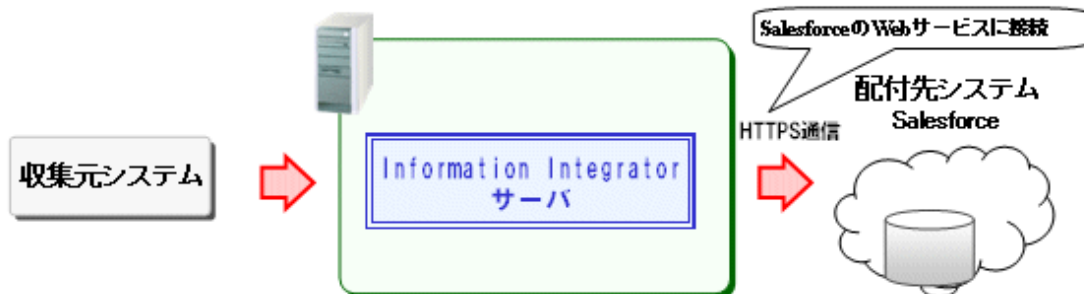


図4.16 配付先システムがSalesforceの場合の構成イメージ



## 4.3.7 帳票データを出力する場合

- ローカルのInterstage List Creatorと連携して帳票出力する場合

図4.17 帳票出力を利用する場合の構成イメージ



利用可能なInterstage List Creatorは、以下のとおりです。

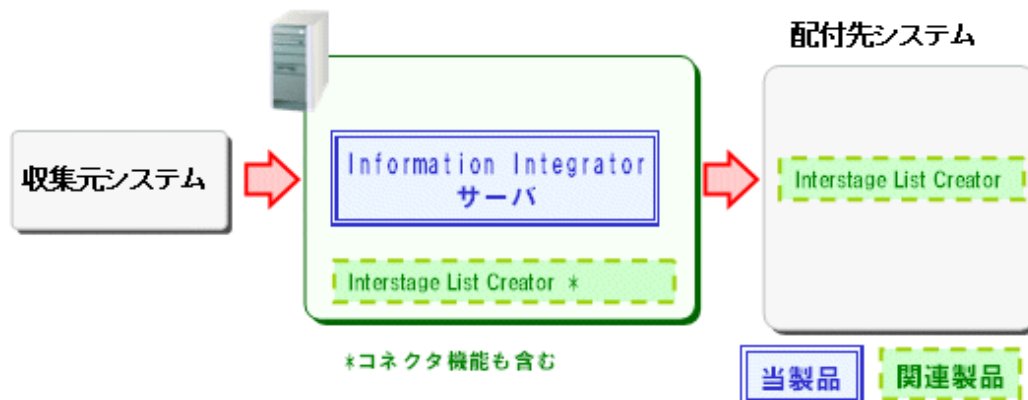
- Interstage List Creator Standard Edition
- Interstage List Creator Enterprise Edition

### P ポイント

ローカルのInterstage List Creatorに接続する場合も、Interstage List Creatorと接続するためのコネクタ機能の設定が必要です。

- リモートのInterstage List Creatorと連携して帳票出力する場合

図4.18 帳票出力を利用する場合の構成イメージ



相手側システムに利用可能なInterstage List Creatorは、以下のとおりです。

- Interstage List Creator Standard Edition
- Interstage List Creator Enterprise Edition

また、相手側システムと接続するため、Information Integratorと同一サーバに、Interstage List Creator Connectorが必要です。

### 4.3.8 相手側システムがInformation Storageの場合

図4.19 収集元システムがInformation Storageの場合の構成イメージ

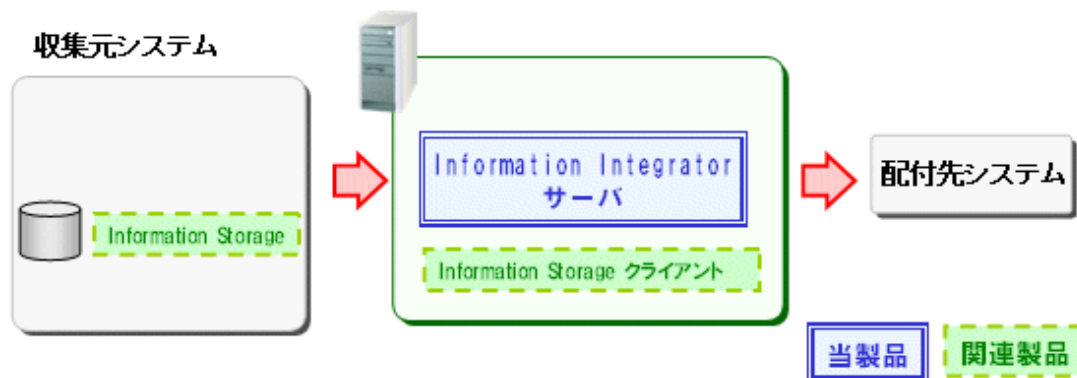
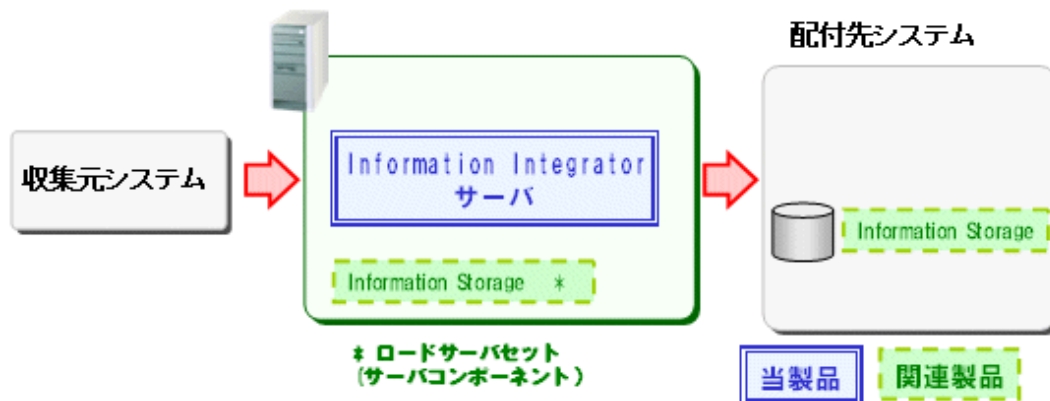


図4.20 配付先システムがInformation Storageの場合の構成イメージ



#### 注意

Information Storageのクライアントを導入後に、Information Storageサーバに接続可能な状態としてください。接続方法については、Information Storageのマニュアルを参照してください。

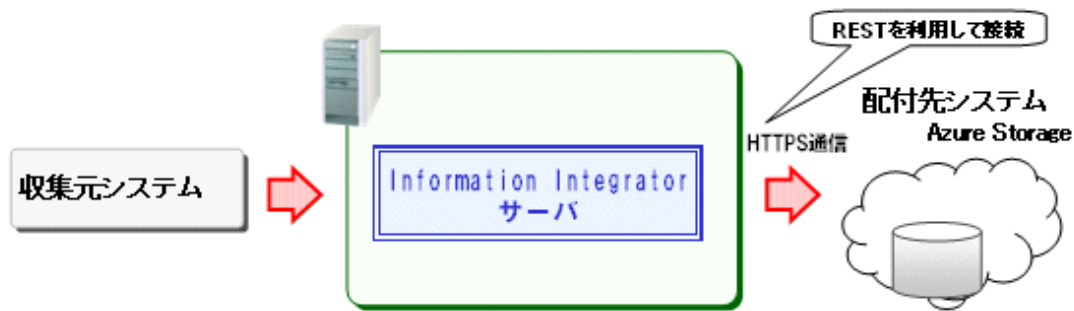
### 4.3.9 相手側システムがAzure Storageの場合

相手側システムがAzure Storageの場合は、特にデータ集配信製品は必要ありません。

図4.21 収集元システムがAzure Storageの場合の構成イメージ



図4.22 配付先システムがAzure Storageの場合の構成イメージ



## 4.4 ファイアウォールを利用する場合

Information Integratorでは、以下の通信間でファイアウォールを利用できます。

- Information Integratorサーバと、Information Integratorクライアントの間
- Information Integratorサーバと、相手側システム(収集元システム、配付先システム)の間

それぞれファイアウォールを利用する場合は、使用するポート番号をファイアウォールに登録してください。使用するポート番号については、“セットアップガイド”を参照してください。

Information Integratorサーバと、相手側システム(収集元システム、配付先システム)の間で、ファイアウォールを利用する場合の留意点は、以下のとおりです。

- 通信プロトコルにFTPを使用した場合  
データコネクション接続モードにより、コネクション確立の方向が異なります。詳細は、“[G.1 FTP/FTP+/FTPMでのコネクション確立シーケンス\(前提知識\)](#)”を参照してください。
- 通信プロトコルにHTTPまたはHTTPSを使用した場合  
通信プロトコルにHTTPまたはHTTPSを使用した場合は、1回の転送に1個のポート番号を使用します。

なお、ファイアウォールのようにポート番号の制限が必要な環境の場合、HTTPまたはHTTPSの使用をお勧めします。

## 4.5 Internet Protocolについて

Information Integratorサーバ、Information Integratorクライアント、相手側システムで、IPv6(Internet Protocol Version 6)、IPv4(Internet Protocol Version 4)のどちらでも動作し、IPv6/IPv4の混在環境でも、動作可能です。

ただし、IPv6/IPv4を相互に異なる環境で利用する場合は、利用できません。

それぞれの環境と、採用されるIPアドレスの関係を以下の表に示します。

表4.4 IPv6、IPv4の環境と、採用されるIPアドレスの関係

Information Integratorサーバ	相手側システム/ Information Integratorクライアント	採用されるIPアドレス
IPv4	IPv4	IPv4
IPv4	IPv6	利用できません。
IPv4	IPv4/IPv6混在環境	IPv4
IPv6	IPv4	利用できません。
IPv6	IPv6	IPv6
IPv6	IPv4/IPv6混在環境	IPv6
IPv4/IPv6混在環境	IPv4	IPv4



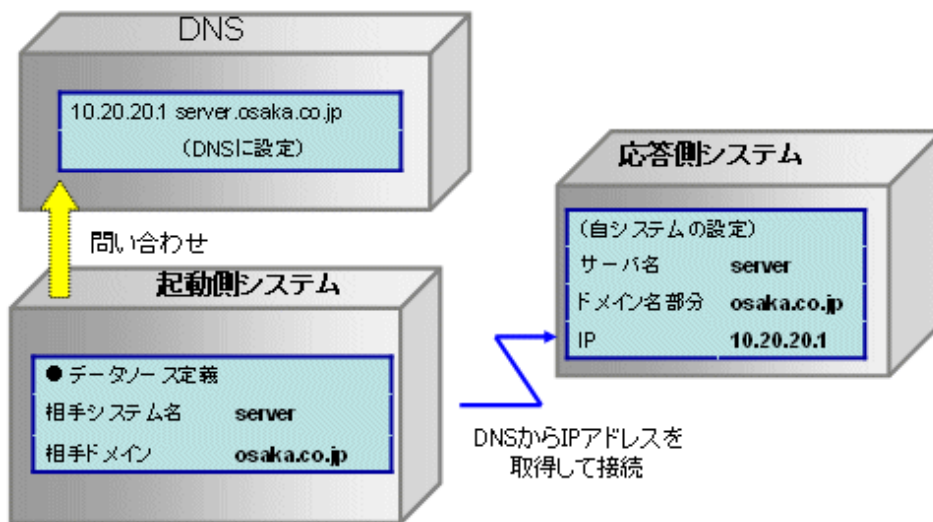
Information Integratorサーバ	相手側システム/ Information Integratorクライアント	採用されるIPアドレス
IPv4/IPv6混在環境	IPv6	IPv6
IPv4/IPv6混在環境	IPv4/IPv6混在環境	自局主導のファイル転送の場合は、Information IntegratorサーバのOSの仕様に、他局主導のファイル転送の場合は、相手側システムのOSの仕様に依存します。

※相手側システムでIPv6を利用する場合は、相手側システムで利用するデータ集配信製品がIPv6に対応している必要があります。データ集配信製品のIPv6への対応可否については、利用するデータ集配信製品のマニュアルを参照してください。

## 4.6 DNSを利用する場合

TCP/IPを利用する場合に完全ドメイン名を利用して相手システムを指定することができます。起動側システムがネットワーク定義に指定された情報を元に応答側システムのIPアドレスを特定する方法を以下に示します。

図4.23 DNSを利用する場合のイメージ



### 完全ドメイン名の記述方法

データソース定義では、以下のオペランドの組合せで完全ドメイン名を表現します。

また、DNSに応答側システムの情報を設定する必要があります。

- 完全ドメイン名の例

```
http://xxxxx.xxx.jp:80
```

- データソース定義

指定項目	指定値
プロトコル	HTTP
相手システム名	xxxxx
相手ドメイン名	xxx.jp
ポート番号	80



## ファイル名の指定方法

以下のURLアドレスを表現する場合、データ収集定義、またはデータ配信定義の「通知ファイル名」に「test/Readme.txt」と指定します。HTTPのみの指定方法です。

```
http://xxxxx.kobe.jp/test/Readme.txt
↑ ↑
(データソース定義) ファイル名の指定
```

## 4.7 相手側システム上にInformation Integratorサーバを導入する場合

Information Integratorサーバを収集元システム、または配信先システム上に、導入することができます。

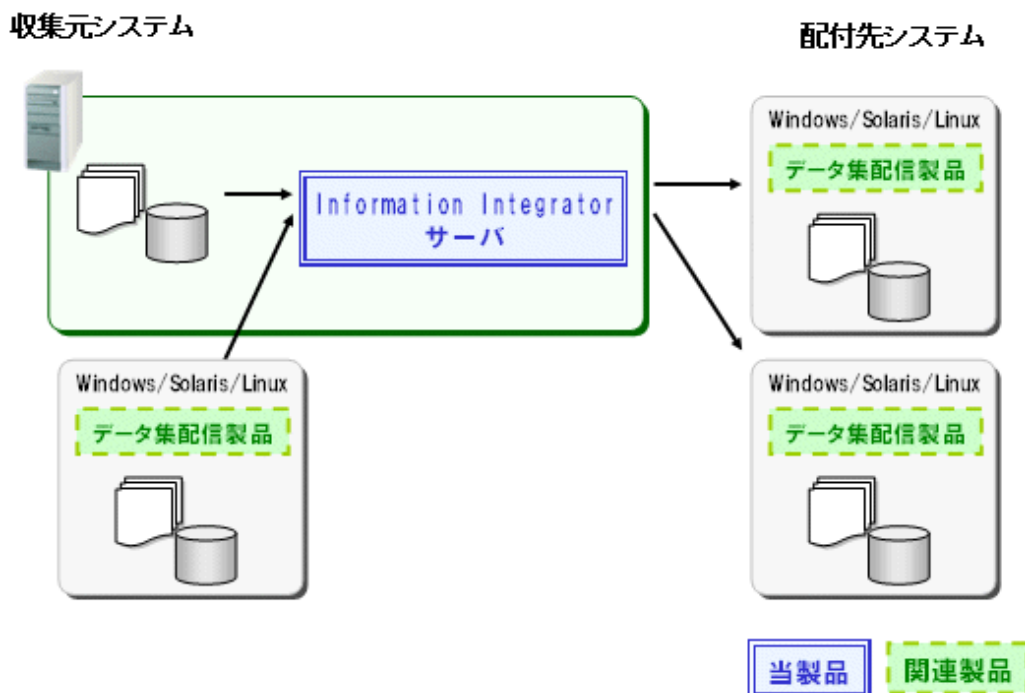
ここでは、収集元システム、または配信先システム上に、Information Integratorサーバを導入する場合のシステム構成について説明します。

導入する前に、相手側システムの業務への影響、システムのスペック、将来拡張の有無などを考慮してから、Information Integratorサーバを導入してください。

### 収集元システム上にInformation Integratorサーバを導入する場合

収集元システム上にInformation Integratorサーバを導入する場合のシステム構成です。Information Integratorサーバを導入していない収集元システムからのデータ収集も通常と同様に可能です。

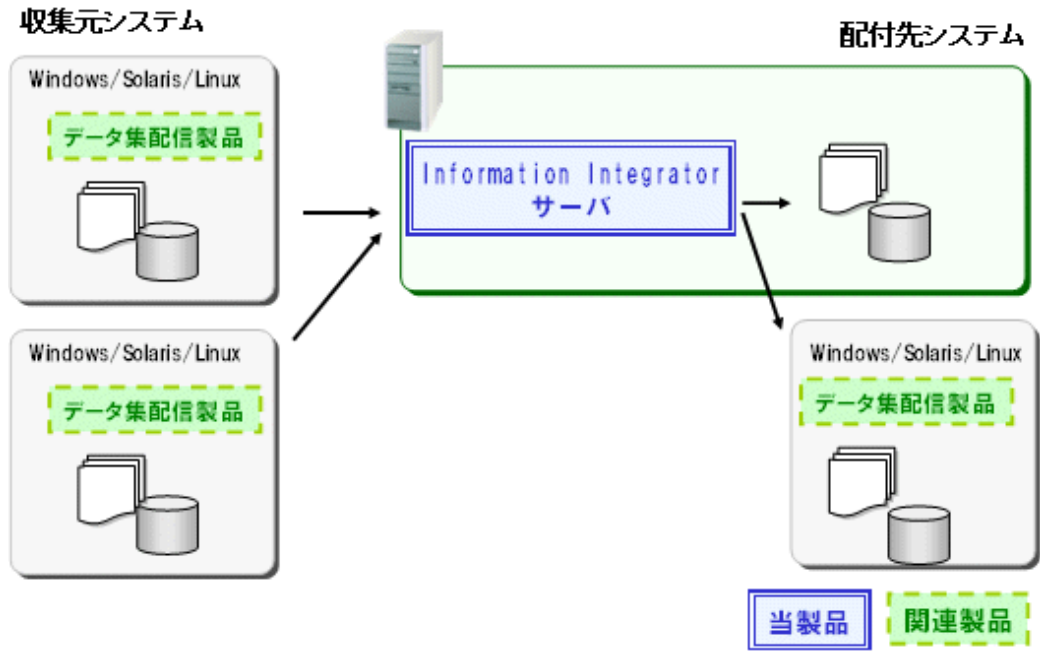
図4.24 収集元システム上にInformation Integratorサーバを導入する場合の構成イメージ



### 配信先システム上にInformation Integratorサーバを導入する場合

配信先システム上にInformation Integratorサーバを導入する場合のシステム構成です。Information Integratorサーバを導入していない配信先システムへのデータ配信も通常と同様に可能です。

図4.25 配付先システム上にInformation Integratorサーバを導入する場合の構成イメージ



## 第5章 処理プロセスの設計

Information Integratorでは、様々な業務システムに存在するデータを収集し、必要なデータ変換処理を行って、最適な形式で配付先システムに配付する一連のデータ処理を処理プロセスと呼びます。

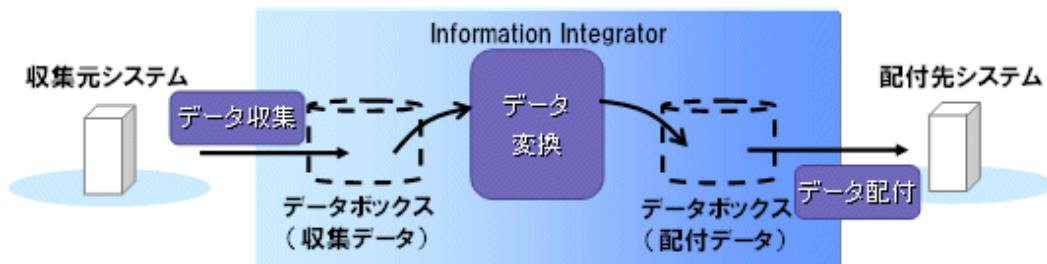
ここでは、Information Integratorでの処理プロセスの設計の考え方と、設計手順を説明します。

### 5.1 処理プロセスの設計の概要

#### 5.1.1 Information Integratorの処理プロセス

Information Integratorの処理プロセスの基本的なイメージを以下に示します。

図5.1 処理プロセスの基本的なイメージ



収集元システムから集めたデータは、Information Integratorのデータ管理機能により、一時的にデータボックスに格納されます。データボックスに格納されたデータに対して、必要な各種変換処理を行い、配付先システムに配付するデータがデータボックス内に作成されます。データボックスに作成された変換処理後のデータを抽出して、配付先システムにデータを配付します。

#### 5.1.2 処理プロセスの設計の流れ

Information Integratorの処理プロセスの設計は以下のように行います。

##### 処理プロセスの設計

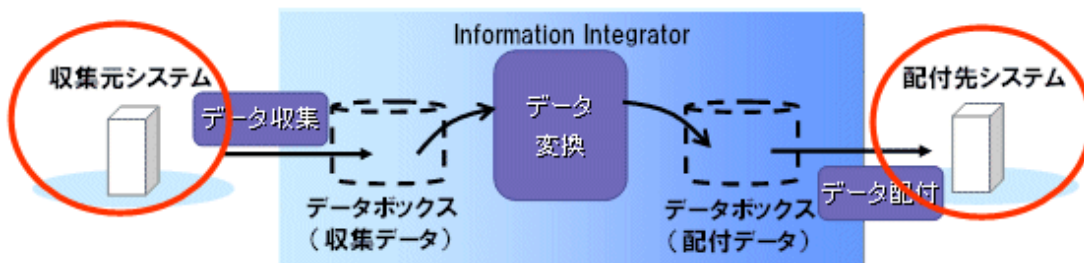
収集処理、変換処理、配付処理の各処理の入出力データの対応付けや処理手順、処理実行スケジュールなど、処理プロセス全体の設計を行います。

##### データの収集、配付処理の設計

1. データの収集元システム、配付先システムの設計

Information Integratorがデータ収集および配付先としてアクセスするシステムの情報(システム名やIPアドレスなど)や、システムへのアクセス手段(データベース情報、ファイル送信のパラメタ、認証情報など)を設計します。

図5.2 処理プロセスに対するデータの収集元システム、配付先システムの設計の対象箇所

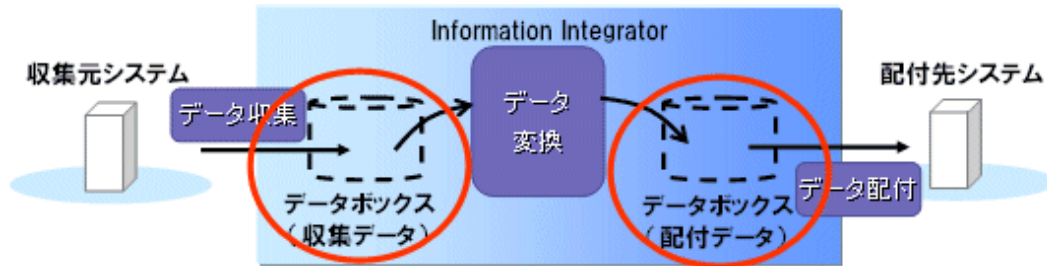


2. 収集データ、配付データの設計

収集元システムより収集するデータ、および配付先システムへ配付するデータについて設計を行います。Information Integratorで処理するデータは、Information Integratorのデータ管理機能を利用し、データボックスに格納して処理を行います。収集データ、配付データを格納するデータボックスは、システムで一意となるデータボックスIDを任意に指定するだけで設計ができます。

収集したデータに対して変換処理を行う場合は、各々のデータに対してデータ構造(項目名やデータ属性など)やフォーマット属性(ファイル形式、文字コードなど)を指定します。

図5.3 処理プロセスに対する収集データ、配信データの設計の対象箇所

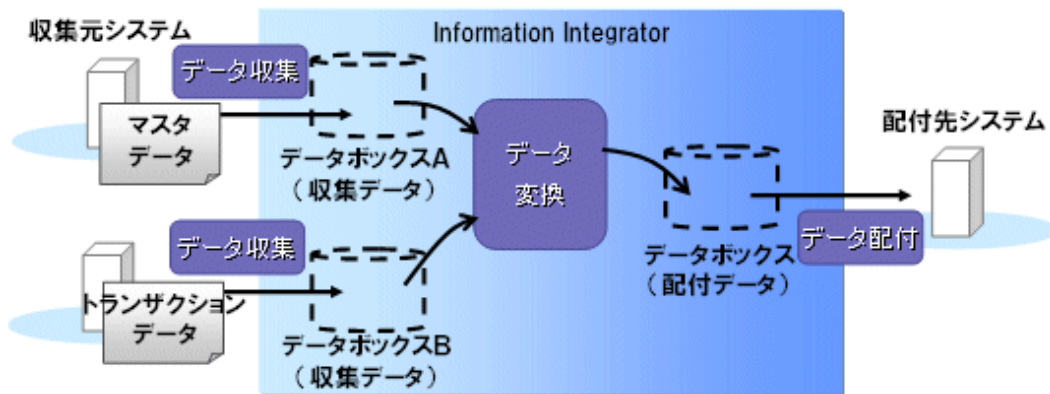


同じデータ構造を持つ1つまたは複数のデータを収集する処理単位を「収集グループ」と呼びます。配信処理も同様に、同じデータ構造を持つ1つまたは複数のデータを配信する処理単位を「配信グループ」と呼びます。

収集データ、配信データを格納するデータボックスは、収集グループおよび配信グループごとに用意します。(データボックスは、Information Integratorサーバのシステム内で一意のIDを指定するだけで設計できます)

例えば、データ構造の異なるマスタデータとトランザクションデータをそれぞれ収集して利用するような場合は、マスタデータとトランザクションデータそれぞれに対して、収集データを格納するためのデータボックスを用意し、収集処理を設計します。

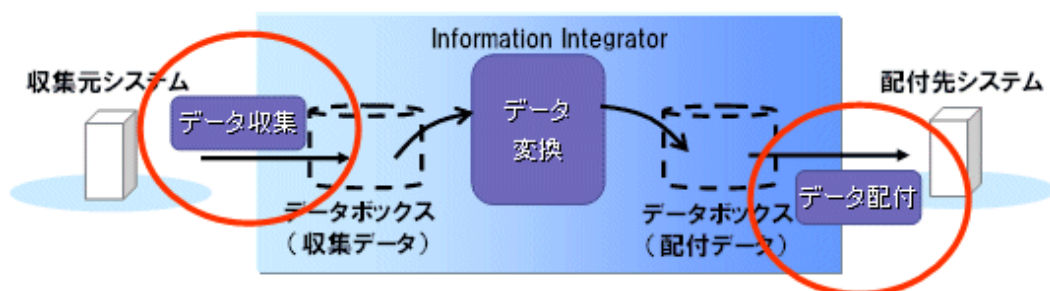
図5.4 データ構造が異なるデータを扱う場合の処理イメージ



### 3. 収集方法、配信方法の設計

収集元システムのデータの形態に合わせて、データベースの抽出やファイルの転送処理に必要な各種パラメタを設計します。配信方法の設計も同様に配信先システムに配信するデータの形態に合わせて設計を行います。

図5.5 処理プロセスに対する収集方法、配信方法の設計の対象箇所

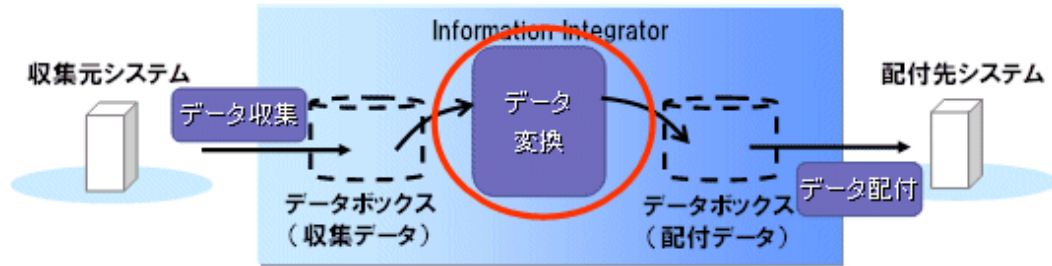


## データの変換処理の設計

収集したデータを配信先システムに配信するデータに変換する処理を設計します。

変換前、変換後のデータ構造やフォーマット属性を設計し、変換前後の項目の対応付けをマッピングすることによって、変換処理を設計します。

図5.6 処理プロセスに対するデータの変換処理の設計の対象箇所



### 外部プログラム呼び出し(プラグイン)の設計

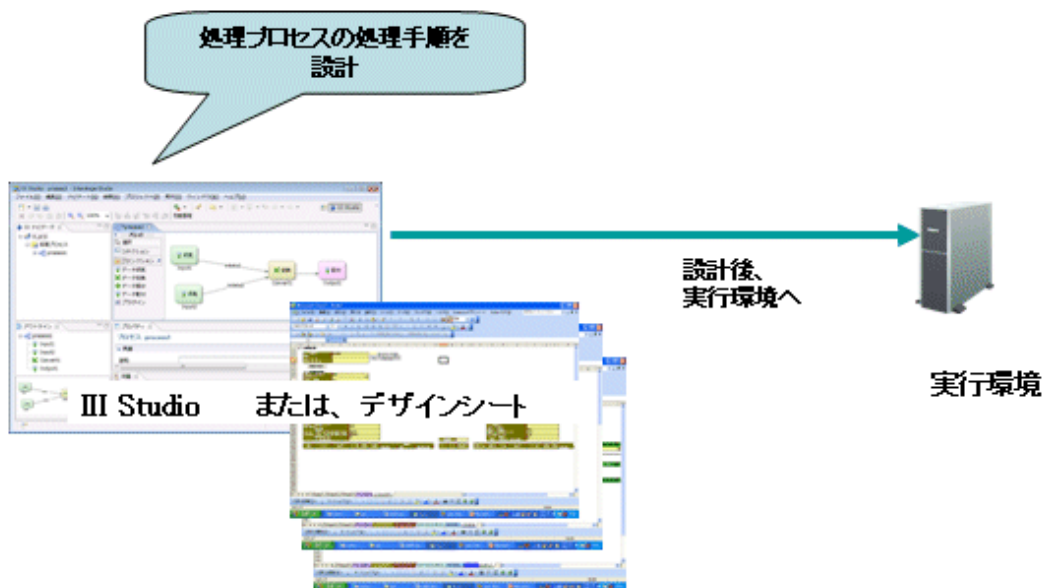
処理プロセスの中で外部プログラムを呼び出す場合には、外部プログラムの情報(コマンド名やパラメタ、復帰値の判定処理など)を設計し、処理プロセスの処理手順にプラグイン呼び出し処理を組み込みます。

## 5.2 処理プロセスの設計作業概要

Information Integratorの処理プロセスを設計する上での基本的な考え方を説明します。III Studioまたは、デザインシートを利用して設計作業を行います。

III Studioを利用すると、処理プロセスの処理手順を視覚的に確認しながら設計を進めることができます。デザインシートを利用すると、類似する複数の処理プロセスを大量に作成するようになるときに、効率的に設計を進めることができます。

設計した処理プロセスをIntegratorサーバへ登録することで、実行環境が作成できます。



III Studioを利用して行う設計作業の詳細は、「第6章 III Studioを利用した設計」を参照してください。デザインシートを利用して行う設計作業の詳細は、「第7章 デザインシートを利用した設計」を参照してください。

## 第6章 III Studioを利用した設計

III Studioを利用した処理プロセスの設計手順について説明します。

処理プロセスの定義をIII Studioで行います。処理プロセスの定義は、以下の定義から構成されています。

- 処理プロセスの処理手順を設計する  
プロセス定義
- 処理プロセスのパラメタを設計する
  - ー 処理プロセスに含めるファンクション処理の定義
    - データ収集定義
    - データ変換定義
    - データ振分定義
    - データ配付定義
  - ー 上記の各ファンクションで必要となるパラメタの設計
    - データ構造定義
    - データソース定義
    - フォーマット定義
    - プラグイン定義

### 6.1 III Studioの起動・停止

#### 6.1.1 III Studioの起動

III Studioの起動方法は、以下のとおりです。



注意

##### 初回起動時の起動方法

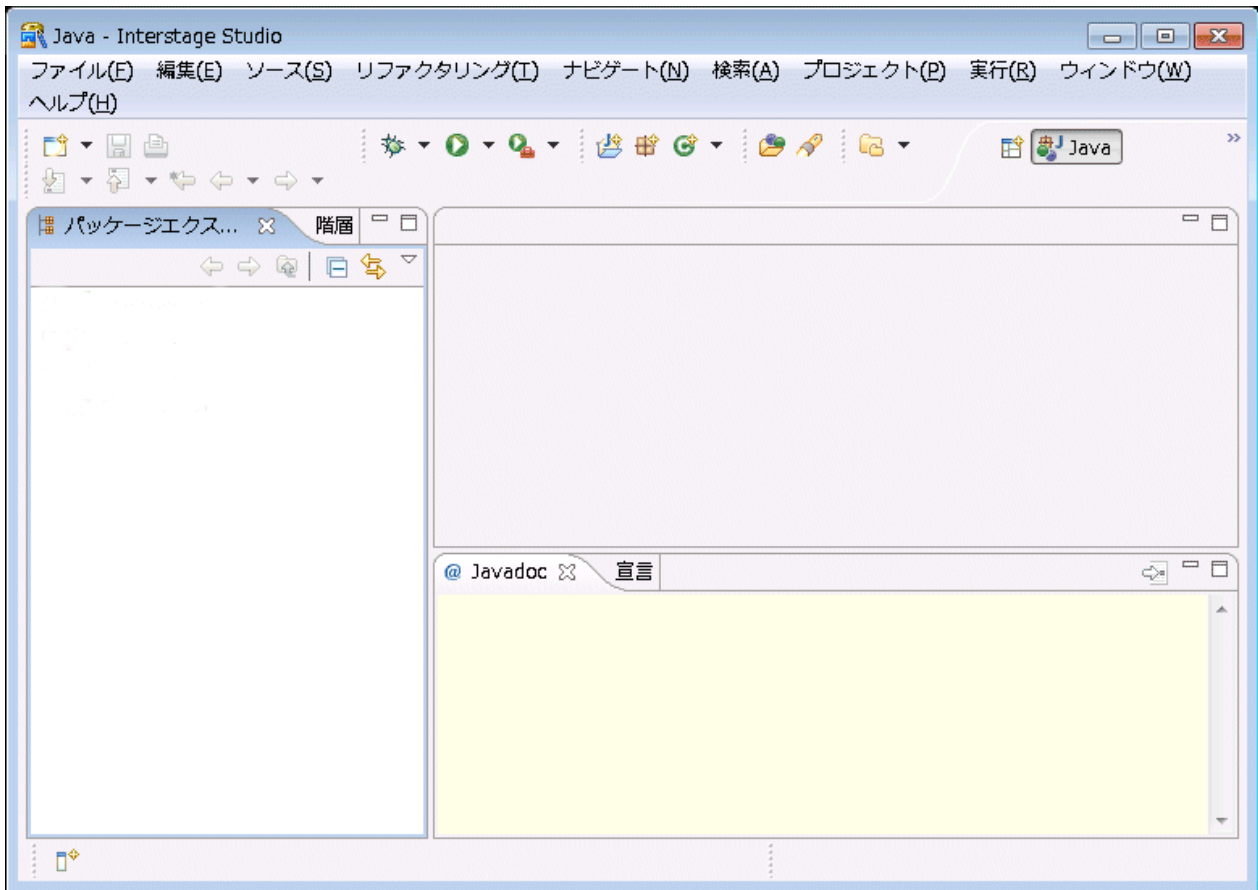
[ファイル名を指定して実行]や[コマンドプロンプト]からInterstage Studioを起動するとき、「-clean」オプションを付与してください。

- Interstage Studio Java EE V9.2の起動コマンドを使用する場合の例

```
<Interstage Studioインストールディレクトリ>%IDE%0902_WB34%eclipse%isstudio.exe -clean
```

III Studioは、GUIベースの統合開発環境Interstage Studio上で動作するプラグインです。スタートメニューから[Interstage Studio]を選択してInterstage Studioを起動します。

→Interstage Studioの初期画面が表示されます。



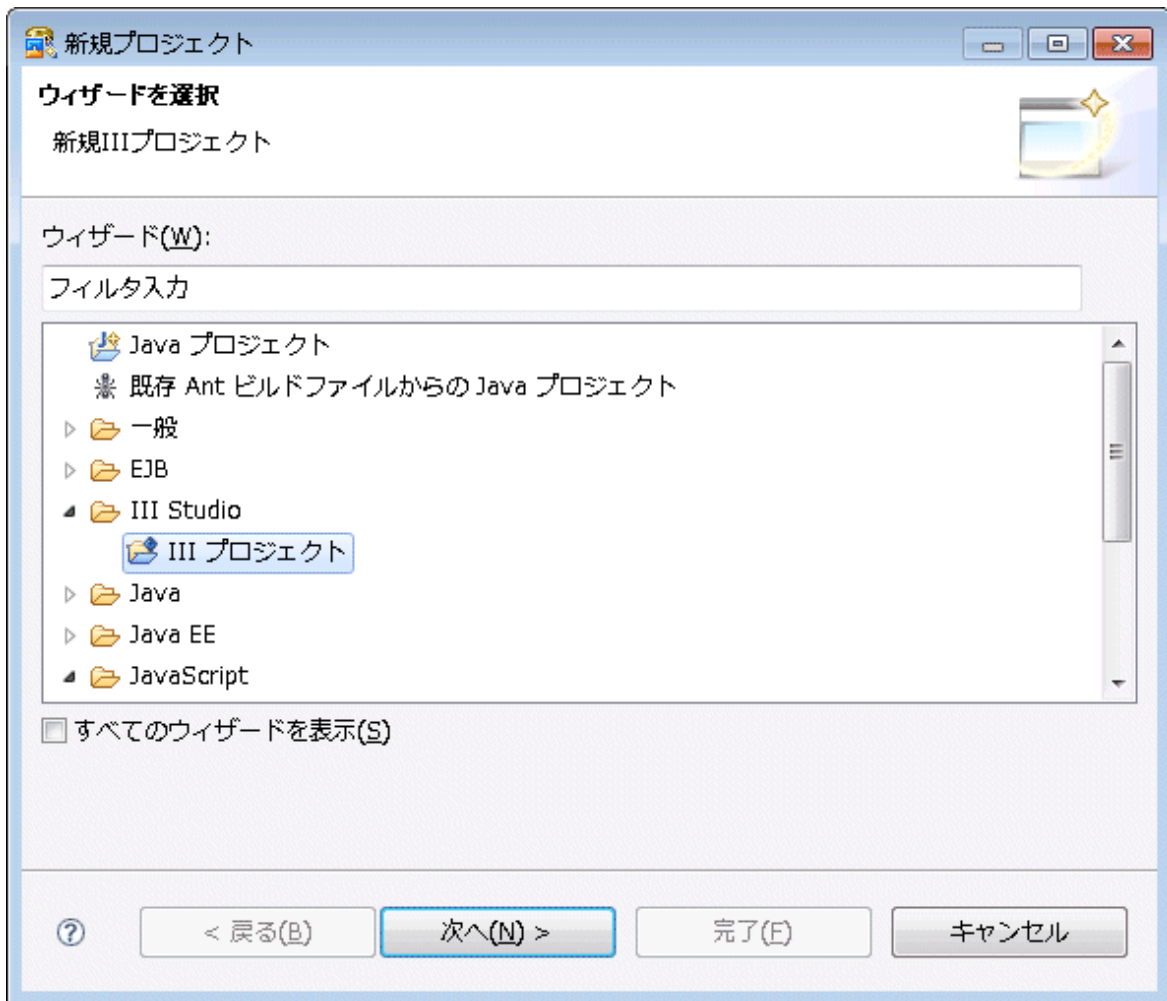
## プロジェクトの新規作成

III Studioを利用するためには、初回利用時に、III Studioで作成した定義を格納する場所(以降、プロジェクトと呼びます)を作成する必要があります。

プロジェクトの作成方法は、以下のとおりです。



1. [ファイル]-[新規作成]-[プロジェクト]をクリックします。  
→[新規プロジェクト]画面が表示されます。





2. [III Studio]-[III プロジェクト]を選択した状態で、「次へ」をクリックします。  
→[新規IIIプロジェクト]画面が表示されます。

新規IIIプロジェクト

IIIプロジェクトを作成します。  
プロジェクト名を指定してください。

プロジェクト名(P):

デフォルトロケーションの使用(D)

ロケーション(L): C:¥Users¥Administrator¥Documents¥Interstage Studi 参照(R)...

? < 戻る(B) 次へ(N) > 完了(F) キャンセル

3. プロジェクト名とプロジェクトの保存先を指定し、[終了]をクリックします。

### 注意

#### プロジェクト名の指定について

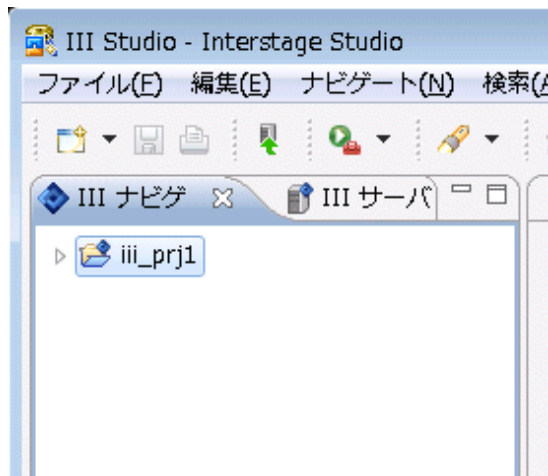
ー 以下の半角記号は指定できません。

- 「"」(ダブルクォーテーション)
- 「\*」(アスタリスク)
- 「/」(スラッシュ)
- 「:」(コロン)
- 「<」(レフトアングル)
- 「>」(ライトアングル)
- 「?」(クエスチョン)
- 「¥」(円マーク)
- 「|」(バーティカル・バー)

ー ピリオドはプロジェクト名の最後尾には使用できません。

→確認メッセージが表示され[はい]をクリックすると、プロジェクトが新規に作成されます。

下記の図は、プロジェクト名「iii\_prj1」を作成した場合です。



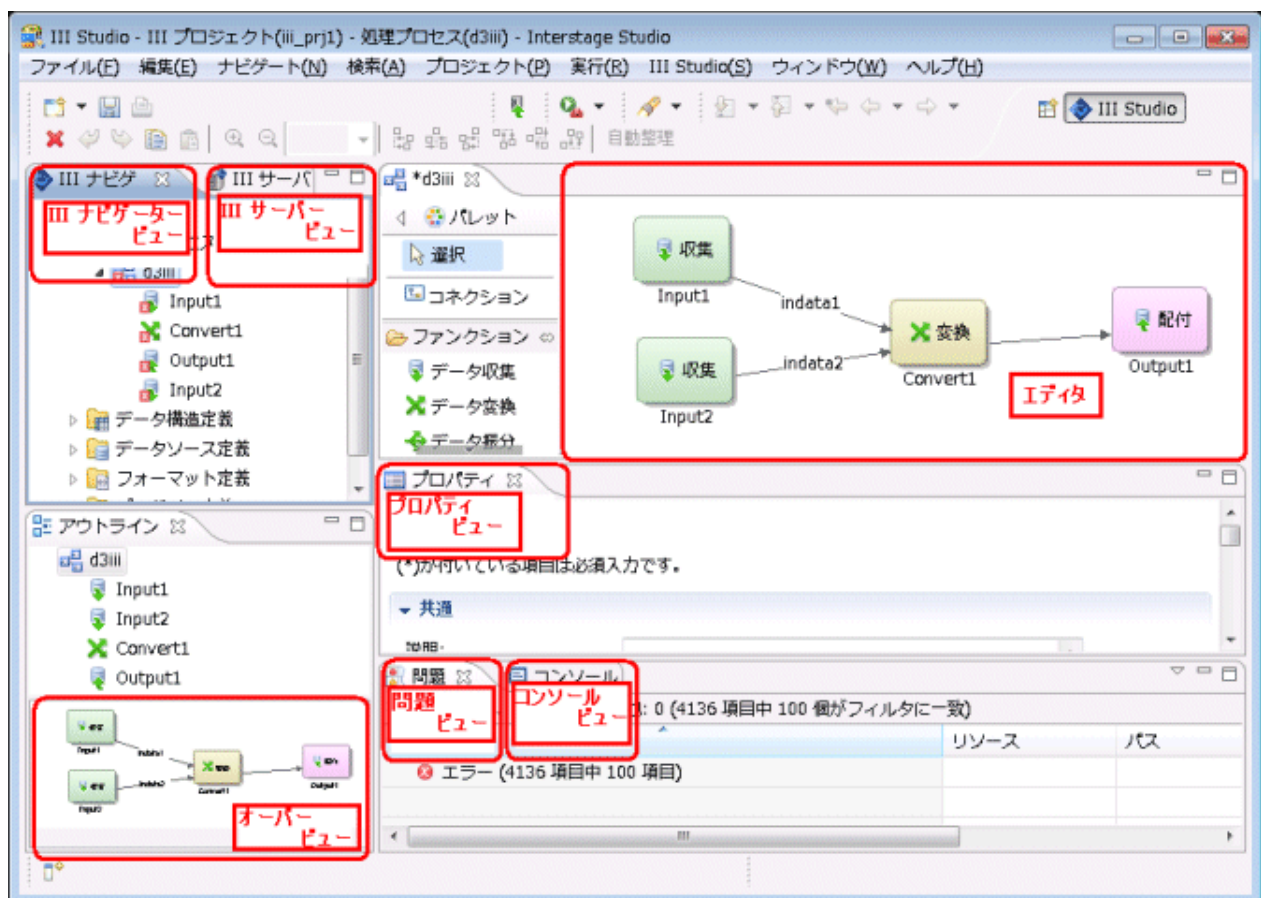
## 6.1.2 III Studioの終了

[ファイル]-[終了]をクリックします。

→[III Studio]画面が閉じます。

## 6.2 III Studioの画面構成

III Studioで処理プロセス設計時に利用する主な画面構成は以下のとおりです。



- **III ナビゲータービュー**  
作成中、または、一時保存された定義をツリー形式で表示します。
- **III サーバービュー**  
Information Integratorサーバに登録されている定義をツリー形式で表示します。
- **エディタ**  
処理プロセスの処理順序や、共有する定義のパラメタを設計します。
- **オーバービュー**  
[III ナビゲータービュー]で処理プロセスを展開すると、作成した定義IDが表示されます。定義IDをダブルクリックすると、[エディタ]に定義の編集内容が表示されます。詳細は、後述の各定義の設計手順を参照してください。
- **IIIプロパティビュー**  
処理プロセスと処理プロセスの各ファンクションのプロパティを表示、設定します。
- **問題ビュー**  
定義内容にエラーがある場合に、エラーの詳細を表示します。
- **コンソールビュー**  
定義情報の登録を行う時、登録の進捗情報を表示します。

III Studioの各項目についての詳細は、“III Studio ヘルプ”を参照してください。

## 6.3 III Studioの利用方法

---

III Studioの利用方法について説明します。

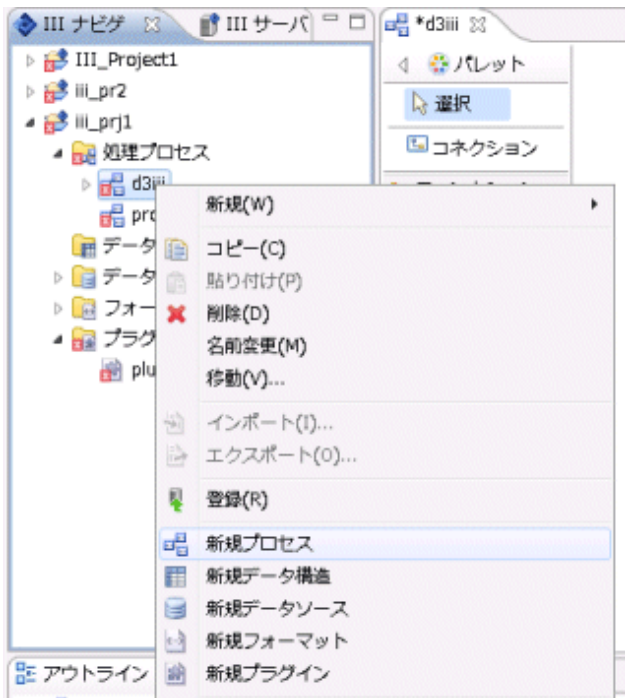
### 6.3.1 定義の新規作成

---

処理プロセスの各定義を新規に作成する手順は、以下のとおりです。

1. [III ナビゲータービュー]の定義を格納するプロジェクト名を選択した状態で、右クリックします。  
→ポップアップメニューが表示されます。

2. ポップアップメニューから、作成する定義に従って選択します。



— [新規プロセス]

処理プロセスの処理手順を設計する(プロセス定義)、および処理プロセスに含めるファンクション処理(データ収集、データ変換、データ振分、データ配付の各定義)の設計を行います。

— [新規データ構造定義]/[新規データソース定義]/[新規フォーマット定義]/[新規プラグイン定義]

処理プロセスに含めるファンクション処理(データ収集、データ変換、データ振分、データ配付の各定義)で必要となるパラメタの設計を行います。

→各定義を管理する定義IDなどを入力する画面が表示されます。

3. 各画面の入力項目に従って、定義IDなどを入力し、最後に[完了]をクリックします。

## ポイント

- 以下の情報は、必ずここで入力してください。

- 各定義ID
- [新規データソース]の「データソースタイプ」
- [新規プラグイン]の[プラグイン種別]

なお、それ以外の入力情報は、後から入力、編集することができます。[プロパティビュー]や[エディタ]で入力、編集することができます。

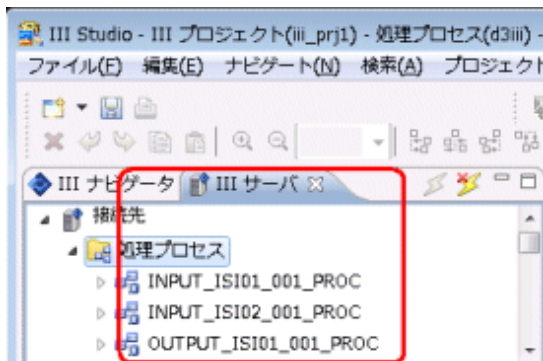
- 各定義情報は、定義ごとに設定する定義ID(プロセス定義ID、ファンクションIDなど)で管理します。定義IDは、Information Integratorサーバのシステム上で一意である必要があります。そのため、各種定義に設定する定義IDが重複しないように、システムで一意になる定義IDを、プレフィックスの付け方などの規約により事前に設計しておくことを推奨します。
- 統合管理オプションとファイル転送エージェントを併用する場合は、運用後に処理プロセス間の関連性を確認できるようにIDを付与してください。例えば、ファイル転送エージェントで利用するプロセス定義IDは、連携するプロセス定義ID(Information Integratorサーバで利用する処理プロセス)に通番を付与するなどしてください。

→[III ナビゲータービュー]に表示されたプロジェクト名を展開すると、作成した定義IDが表示されます。定義IDをダブルクリックすると、[エディタ]に定義の編集内容が表示されます。詳細は、後述の各定義の設計手順を参照してください。

## 6.3.2 定義をサーバに登録

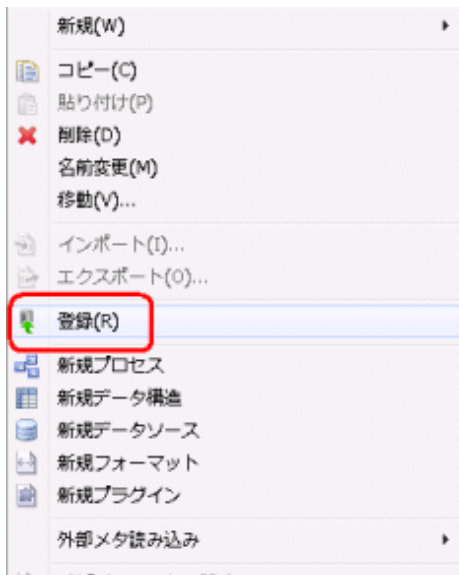
1. Information Integratorサーバに接続します。[III Studio]-[サーバ接続]を選択します。

→[IIIサーバービュー]に接続先のサーバが表示されます。展開すると、登録されている定義が表示されます。



2. [IIIナビゲータービュー]で登録する定義を選択した状態で、右クリックします。

→ポップアップメニューが表示されます。

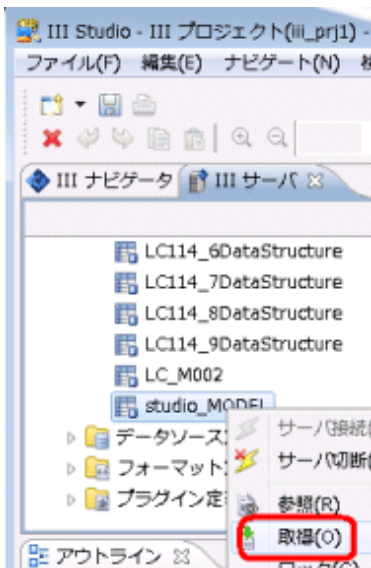


3. 「登録」を選択します。  
→Information Integratorサーバに定義が登録されます。
4. Information Integratorサーバから切断します。[III Studio]-[サーバ切断]を選択します。  
→[IIIサーバービュー]に表示されていた定義情報が表示されなくなります。

## 6.3.3 サーバに登録した定義の修正方法

1. [IIIサーバービュー]で修正する定義を選択した状態で、右クリックします。なお、複数定義を選択することも可能です。  
→ポップアップメニューが表示されます。

2. 「取得」を選択します。



→定義を格納するプロジェクトを選択すると、[IIIナビゲータービュー]の当該プロジェクト配下に定義が格納されます。定義の修正が可能になります。

### ポイント

定義を取得後、何らかの理由により編集を取りやめる場合は、取得の取り消し操作が必要です。[IIIサーバービュー]で変更対象を選択した状態で、右クリックメニューから[ロック解除]を選択してください。なお、本操作は、当該定義を取得したユーザーまたは、システム管理者だけ行うことができます。

## 6.3.4 定義の一時保存・再編集

- ・ 保存方法  
[ファイル]-[保存]をクリックします。  
→定義の情報が、III Studioが導入されているPC上に保存されます。
- ・ 再編集方法  
[IIIナビゲータービュー]に表示されているツリービューを展開し、再編集を開始する定義をダブルクリックします。  
→[エディタ]に対象の定義が表示されます。

## 6.3.5 定義の複写

1. [IIIナビゲータービュー]で複写する定義IDを選択した状態で、右クリックします。  
→右クリックメニューが表示されます。[コピー]を選択します。
2. 手順1と同様に、右クリックメニューから[貼り付け]を選択します。  
→定義IDを編集する画面が表示されます。定義IDを入力して[OK]をクリックします。定義が複写されます。[IIIナビゲータービュー]に、複写した定義IDが表示されます。

複写した定義IDを変更し、定義内容を編集して登録することで、同じ定義の設計を繰り返し行うことなく定義作成が可能になります。

## 6.4 処理プロセスの処理手順の設計

処理プロセスの処理手順の設計について説明します。

設計手順は、以下のとおりです。

## プロパティを確認する

処理プロセスをエディタで開き、ファンクションやコネクションの配置されていない領域を選択すると、[プロパティビュー]に処理プロセスのプロパティが表示されます。

また、エディタ上でファンクションを選択すると、[プロパティビュー]にファンクションのプロパティが表示されます。

- ・ 処理プロセス全体に対する設定項目の指定方法

プロセスグループIDや実行スケジュールなど、処理プロセス全体に対する設定項目の指定方法について説明します。

項番	ファンクションID*	処理の種類*	変換グループID	出力対象とする	データボックス	例外処理ID	リトライ実施	リトライ回数	リトラ

### a. プロセスグループの指定

ユーザー設計に基づいて、プロセスグループを指定します。

### b. 順序性制御

同一の処理プロセスについて順序性制御を行うか否かを指定します。

### c. 処理プロセスの実行スケジュール

処理プロセスの実行スケジュールを設定する場合、実行スケジュールの種別と開始時刻を設定します。

外部アプリケーションからコマンドでプロセスを起動する場合は、設定しません。

## ポイント

一定時間間隔繰り返しとイベント(収集)監視実行では、スケジュール実行時間帯を事前に設定しておくことができます。例えば、Information Integratorに、60分間隔で繰り返し実行するプロセスや、収集監視実行のプロセスなどがあり、いずれのプロセスも実行開始時間帯を午前9時以降午後5時までと設定しておくことができます。

この設定は、動作環境ファイルに行います。本設定についての詳細は、“セットアップガイド”を参照してください。

## 例

イベント監視を行う場合



設定項目	設定内容
実行スケジュール種別	イベント監視

なお、1つの処理プロセスに、イベント監視対象データと監視対象外データを混在させることはできません。イベント監視を選択した場合は、必ずデータ収集定義の「収集監視」に「あり」と定義してください。

各項目の詳細は、“III Studioヘルプ”を参照してください。

また、イベント監視を行う場合のデータ収集定義の設定内容については、“6.6.3 収集方法の設計”を参照してください。

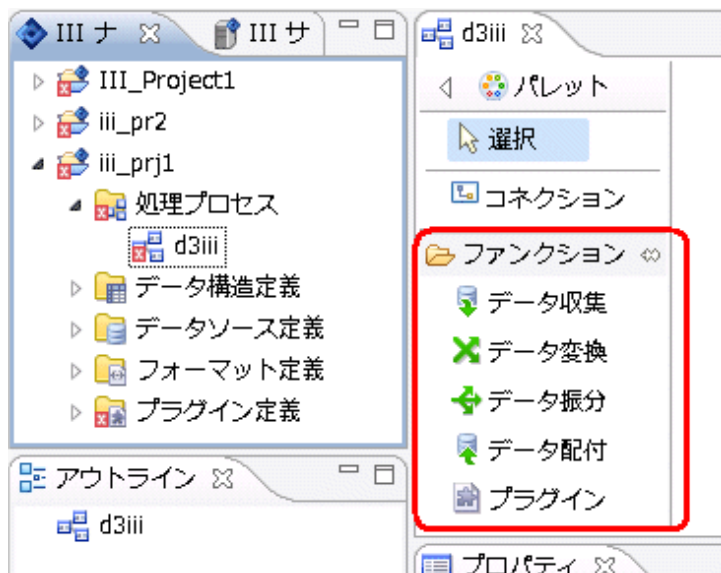
#### d. リカバリポイント

リカバリポイントを使用するか否かを指定します。

### 処理手順を設計する

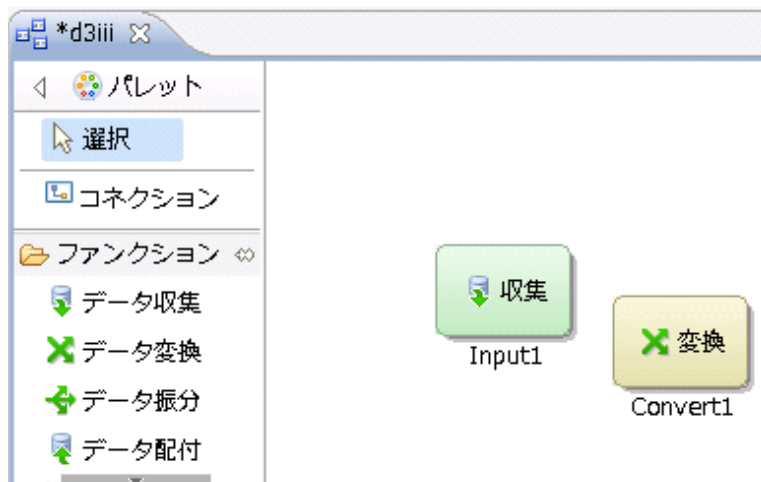
エディタで処理プロセスの処理手順を設計します。操作方法は、以下のとおりです。

1. [III ナビゲータービュー]で、設計する処理プロセスをダブルクリックします。  
→新しいタブがエディタに追加され、選択した処理プロセスの編集領域が表示されます。
2. ファンクションを追加します。パレットから「ファンクション」を選択し、キャンバスへドラッグします。



→キャンバス上にファンクションが作成されます。定義IDは、必要に応じて変更します。

下記の図は、データ収集およびデータ変換のファンクションを1つずつ作成した例です。

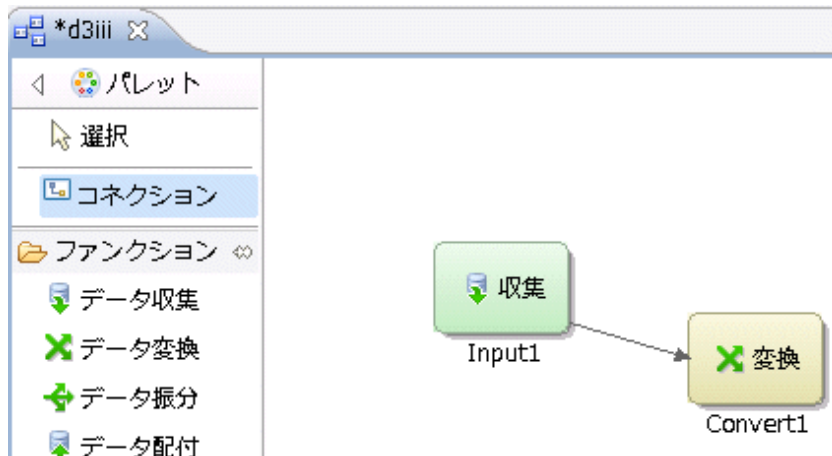




複数のファンクションを作成する場合は、本操作を必要な数だけ繰り返します。

3. ファンクションの処理順序を指定します。パレットから「コネクション」を選択し、キャンバス上に追加した先行処理と後処理に該当するファンクションを順番にクリックします。

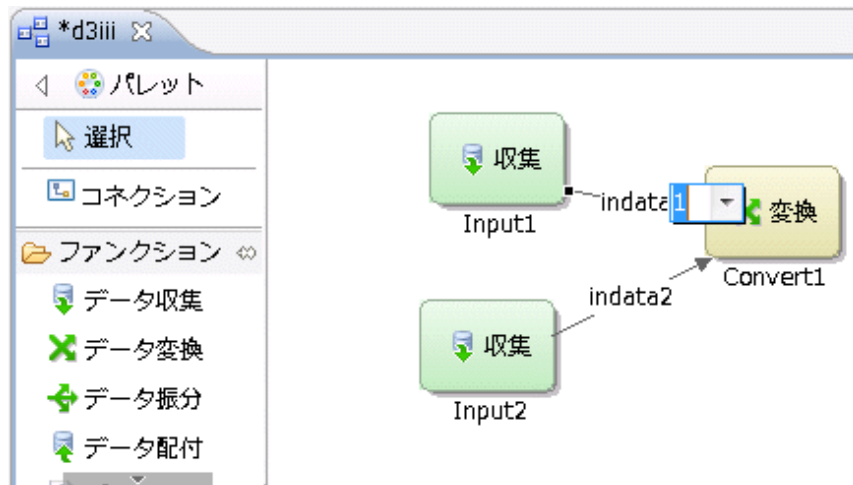
→選択したファンクション間に処理順序を示す矢印が表示されます。



なお、入出力データが複数ある場合は矢印の上にデータIDが表示されます。複数のファンクションの処理順序を指定する場合は、本操作を必要な数だけ繰り返します。

データIDについての詳細は、「6.4.1 ファンクション処理間のデータ受け渡しの設計」を参照してください。

4. 必要に応じてデータIDを変更します。パレットから「選択」を選び、データIDが表示されている矢印を一度選択した状態にしてから、再度矢印をクリックします。



→IDを指定します。

- － 結びつけの後続処理がデータ変換定義の場合

データ変換定義の入力データに対するID(変換データID)を「indataN(N:1からの整数)」の形式で指定します。

- － 結びつけの先行処理がデータ振分定義の場合

データ振分定義の出力データに対するID(振分データID)を「OutDataN (N:1から100までの整数)」の形式で指定します。

5. III Studioで作成した処理プロセスの定義内容にエラーがないことを確認します。

→エラーまたは警告がある場合は、[III問題ビュー]に情報が表示されます。エラーまたは、警告が表示されている場合は、表示内容に従って対処してください。

6. 設計情報を保存します。[ファイル]-[保存]をクリックします。

→処理プロセスの情報が保存されます。

## 注意

- データ収集およびデータ配付処理を必ず作成してください。
- プロセス定義の先頭処理には、データ収集を指定し、最終処理には、データ配付を指定してください。ただし、処理プラグイン機能を用いた処理については、先頭および最終の処理として指定することができます。
- 処理プラグイン機能を用いた処理を先頭の処理に指定する場合、後続処理に、対応するデータ収集処理を必ず指定してください。また、その場合、後続のデータ収集処理が複数存在する場合は、プラグイン処理を、データ収集処理の数分用意してください。

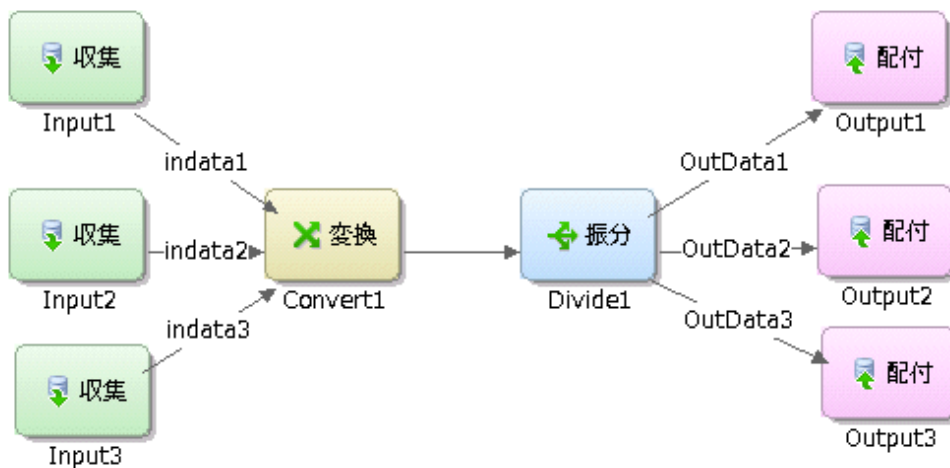
### 6.4.1 ファンクション処理間のデータ受け渡しの設計

データ変換処理の入力データ(変換前のデータ)や、データ振分処理の出力結果が複数のデータになる場合に、ファンクション処理の間で受け渡すデータの対応関係を設計する必要があります。

データ変換処理の入力データや、データ振分処理の出力データには、設計時に自動でデータIDが設定されます。(「変換データID」および「データ振分定義ID」)

ファンクション処理で扱うデータの対応付けは、プロセス定義において、「変換データID」および「振分データID」を前後のファンクションに対して指定することにより設計します。

以下に入力例を示します。



上記は以下の対応付けを行う例です。

- データ収集処理「Input1」で収集したデータを、データ変換処理「Convert1」の入力データindata1に対応付け
- データ収集処理「Input2」で収集したデータを、データ変換処理「Convert1」の入力データindata2に対応付け
- データ収集処理「Input3」で収集したデータを、データ変換処理「Convert1」の入力データindata3に対応付け
- データ振分処理「Divide1」で振り分けたデータOutData1を、データ配付処理「Output1」の配付データに対応付け
- データ振分処理「Divide1」で振り分けたデータOutData2を、データ配付処理「Output2」の配付データに対応付け
- データ振分処理「Divide1」で振り分けたデータOutData3を、データ配付処理「Output3」の配付データに対応付け

## 注意

- 複数のデータ振分処理を先行処理として選択することはできません。
  - データ振分処理の後続処理が異常完了した場合、データ振分処理が再開待ち状態になります。データ振分処理をiffrstrコマンドで再開してください。(プロセスの先頭からの再実行はできません。)
- コマンドの詳細は、“コマンドリファレンス”を参照してください。

- データ振分処理の後続処理として、Service Integratorのサービスバス連携出力が含まれるプロセス処理においては、データ振分処理を作成しないでください。

## 6.5 処理プロセスのパラメタの設計

処理プロセスのパラメタの設計を行います。

1つの処理プロセスの中で実行する、収集処理、変換処理、配付処理の各処理の単位を、ファンクション処理と呼びます。これらのファンクション処理の定義を行います。

以下にファンクション処理の種類と、ファンクション処理の処理単位を以下に説明します。

表6.1 ファンクション処理の種類

ファンクション処理の種類	1つのファンクション処理となる処理単位
データ収集処理	同じデータ構造の1つまたは複数データを収集してデータボックスに格納する処理(収集グループ)。
データ変換処理	1つまたは複数のデータを入力データとして変換処理を行い、1つのデータを出力する処理。
データ振分処理	1つのデータを入力データとし、振分条件によってレコードごとに複数データに振り分ける処理。
データ配付処理	データボックスに格納されたデータを、1つ以上の配付先システムに配付する処理(配付グループ)。
プラグイン処理	1つのプラグイン(外部プログラム)を呼び出す処理。

処理プロセスに含める各ファンクション処理を、それぞれ定義します。

### 6.5.1 処理プロセスのパラメタを設計する定義種別

処理プロセスのパラメタを設計するための定義は、以下のとおりです。

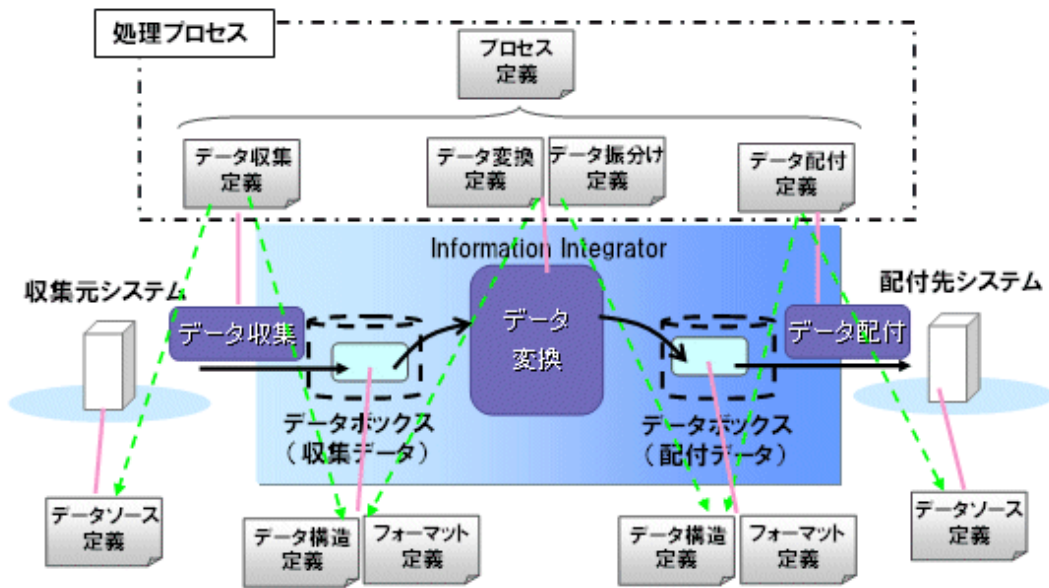
表6.2 処理プロセスのパラメタを設計するための定義

定義名	定義概要
データ収集定義	データ収集処理に対して、収集対象とするデータの所在、収集手段を定義します。
データ変換定義	データ変換の処理内容(フォーマット変換、項目編集、レコード編集)を定義します。 変換前、変換後のデータに対して、データ構造とフォーマットを定義し、項目の対応付け(マッピング)を定義します。また、演算式を利用した項目ごとの編集、集計やソート、ジョインなどのレコード全体に対する編集を定義します。
データ振分定義	データを、振分条件によってレコード毎に複数データに振り分ける処理を定義します。
データ配付定義	データ配付処理に対して、配付先の所在、配付手段を定義します。
データソース定義	収集先、および配付先のシステム上にあるデータを収集、配付処理を行う上で必要な基本情報(システムの情報、アクセス手段など)を定義します。 データ収集定義、データ配付定義からIDで参照する情報です。
フォーマット定義	データ変換を行う際に、変換前と変換後のデータのフォーマットとして、文字コードやデータ書式などを定義します。
データ構造定義	データ変換を行う際の、変換前と変換後のデータの構造として、項目名やデータ属性、桁数などを定義します。
プラグイン定義	データ処理の中で呼び出すプラグイン(外部プログラム)の情報を定義します。

処理プロセスに対して、設計する範囲と各定義の対応付けを以下の図に示します。(プラグイン定義は、図中にありませんが、処理プロセスでプラグイン呼び出しを行う場合に作成します)

点線矢印は、定義間の参照関係を示しています。(例えば、収集元システムに対するデータソース定義の情報は、データ収集定義から参照します。)

図6.1 設計する範囲と各定義の対応付け

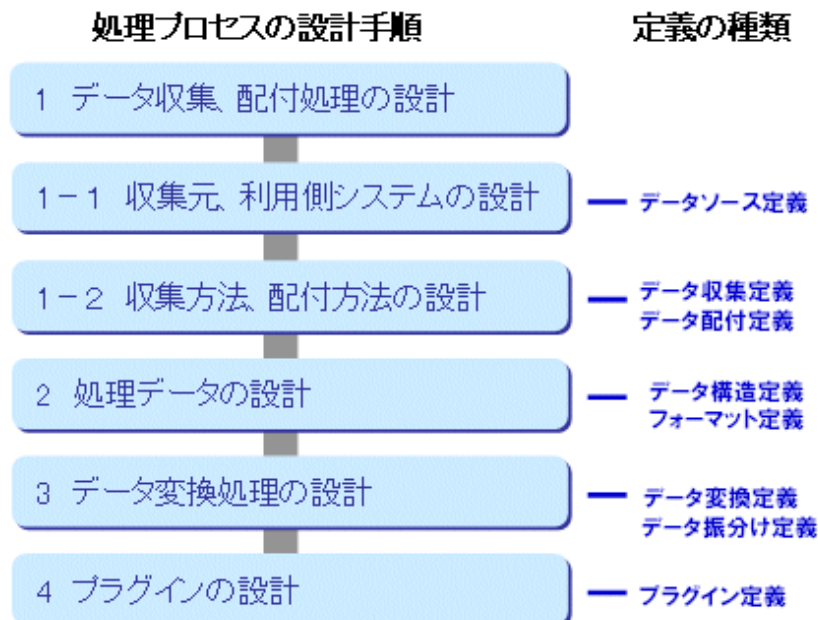


プロセス定義は、処理プロセスの処理手順の設計を行います。詳細は、“6.4 処理プロセスの処理手順の設計”を参照してください。

## 6.5.2 処理プロセスのパラメタの設計手順

各定義を作成し、処理プロセスのパラメタの設計を行います。設計の流れと、各処理の設計時に利用する定義を以下に示します。

図6.2 処理プロセスのパラメタの設計手順



各設計手順については、後述を参照してください。

## 6.5.3 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、以下の留意事項があります。

- ・ 見出し行が存在するデータで、データ変換処理またはデータ振分処理を行う場合
  - 変換対象のデータを事前に収集するときに(データ収集時)、必ず収集対象のデータが、個別にデータボックスに格納されるようにしてください。  
データ収集定義では、データボックスへの格納方法で「個別型」を指定します。
  - 変換対象のデータが一般フォーマットのデータ構造の場合のみ、データ変換処理またはデータ振分処理が可能です。  
データ構造定義では、変換前データの「データ形式」に「一般」を指定します。なお、一般フォーマットについての詳細は、“[付録D フォーマット形式](#)”を参照してください。
  - 横連結のジョイン(INNER JOIN、OUTER JOIN)は、マスタデータに見出し行が存在するデータは指定できません。  
変換前データの「indata2」以降で設定するフォーマットIDは、フォーマット定義の見出し行の有無に「なし」を指定した場合のみ利用可能です。
  - 縦連結のユニオン (UNION、UNION ALL)は、できません。
  - 変換前データからレコードを抽出する機能(レコード抽出機能)は、できません。
- ・ データ変換処理またはデータ振分処理後に見出し行を付加する場合
  - 変換対象のデータが一般フォーマットのデータ構造の場合のみ、データ変換処理またはデータ振分処理が可能です。  
データ構造定義では、変換後データの「データ形式」に「一般」を指定します。なお、一般フォーマットについての詳細は、“[付録D フォーマット形式](#)”を参照してください。
  - 項目名を元に見出し行を生成します。変換後の文字コードの指定によっては、正しく変換/出力されない場合があります。

III Studioについての詳細は、III Studioの各項目に設定されたツールチップ、および“[III Studio ヘルプ](#)”を参照してください。

## 6.6 データ収集処理の設計

データ収集処理の設計手順について説明します。

データ収集処理の設計は以下の手順で行います。

1. 収集元システムの設計
2. 収集データの設計
3. 収集方法の設計

### 6.6.1 収集元システムの設計

データの収集元としてアクセスするシステムの情報(システム名やIPアドレスなど)や、システムへのアクセス手段(データベース情報、ファイル受信のパラメタ、認証情報など)を設計します。

#### データソース定義の作成

[III ナビゲータービュー]でデータソース定義を展開し、設計するデータソース定義IDをダブルクリックします。[エディタ]にデータソース定義の各項目が表示されます。ここで、データソース定義の作成を行います。

#### ポイント

データソースタイプによって、エディタに表示される項目が異なります。データソースタイプは、データソース定義を新規に作成する場合に選択します。詳細は、“[6.3.1 定義の新規作成](#)”を参照してください。

- ・ 収集元システムのデータベースからデータを抽出する場合  
収集元システムのデータベースに関するパラメタを入力します。
- ・ 収集元システムのデータをファイル受信で収集する場合  
収集元システムやファイル受信に必要なパラメタを入力します。

- ・ 収集元システムのSalesforceからデータを抽出する場合  
収集元システムやオブジェクト抽出に必要なパラメータを入力します。
- ・ 収集元システムのInformation Storageからデータを抽出する場合  
収集元システムやIIS抽出に必要なパラメータを入力します。
- ・ 収集元システムのAzure StorageからBLOBデータを収集する場合  
収集元システムやBLOB収集に必要なパラメータを入力します。
- ・ 収集元システムのService Integratorからデータを収集する場合  
収集元システムやデータ収集に必要なパラメータを入力します。

## ポイント

### Service Integratorからデータを収集する場合

サービスバス連携入力を行う場合は、Service Integrator側の設定が必要です。Service Integratorの設定についての詳細は、“[付録 J サービスバス連携を行う場合のService Integratorの設定](#)”を参照してください。

他の処理プロセスのデータ配付処理や、Information Integratorのデータボックス格納コマンドを利用して、データボックスに格納したデータを利用して処理を行う場合は、データ収集定義の「共通」部分にのみパラメータを入力します。

## ポイント

サーバに登録済みのデータソース定義がある場合、データ収集定義の「データソースID」に、登録済みのデータソース定義の定義IDを指定して、参照利用することが可能です。同じデータソース定義を再作成する必要はありません。

## 6.6.2 収集データの設計

### データボックス名の作成

収集したデータを格納するデータボックス名を決めます。Information Integratorサーバでデータを収集して処理するためには、データボックスが必要です。データボックス名の入力は、[プロパティビュー]で行います。[エディタ]で作成するデータ収集定義を選択すると、[プロパティビュー]にデータ収集定義の各項目が表示されます。表示された項目のうち、[共通]-[データボックス]にデータボックス名を入力します。

データボックス名は、処理プロセス内で一意の値を設定する必要があります。複数プロセスに同じデータボックス名を設定すると、複数プロセス間でデータを受け渡すことができます。

### 収集データのデータ構造、フォーマットの設計

収集してデータボックスに格納するデータのデータ構造、フォーマットを、それぞれデータ構造定義、フォーマット定義で設計します。

## ポイント

以下の両方を満たす場合、収集データの構造やフォーマットの設計は不要です。

- ・ ファイル形式またはサービスバス連携形式のデータ収集の場合
- ・ 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、フォーマットの設計は不要です。

- ・ オブジェクト抽出によるデータ収集の場合
- ・ BLOB収集によるデータ収集の場合



## 6.6.3 収集方法の設計

収集方法の設計は、データ収集定義を利用して行います。以下に設計手順を説明します。

### データ収集定義の作成

[III ナビゲータービュー]で処理プロセスを展開し、設計する処理プロセス定義IDをダブルクリックします。[エディタ]に処理プロセスの処理手順が表示されます。[エディタ]で作成するデータ収集定義を選択すると、[プロパティビュー]に収集定義の各項目が表示されます。ここで、データ収集定義を作成します。



例

#### イベント監視を行う場合

設定項目	設定内容
運用種別	応答側
収集監視有無	あり
0件データの扱い	正常 イベントを待ち合わせない場合に指定します。
	異常 イベントを待ち合わせる場合に指定します。
収集ファイル名	<p>完全一致、あるいはワイルドカード(*)により指定します。 ディレクトリ監視の場合のみ指定可能です。</p> <ul style="list-style-type: none"> <li>ワイルドカード指定で、収集するファイル名が「監視対象ディレクトリ名+退避先ディレクトリ名+ファイル名の合計が230バイト」以上の場合 処理ファイル名の指定値に合致していても、常に処理対象外となります。</li> </ul> <p>また、処理プロセスの実行対象外となるファイルの扱いは、動作環境ファイルの「monitorexcept」パラメタの指定に関わらず、常に削除されます。</p> <p>なお、収集するファイル名が「監視対象ディレクトリ名+退避先ディレクトリ名+ファイル名の合計が230バイト」未満の場合は、ワイルドカード指定の指定値に合致したものを処理対象として動作します。処理プロセスの実行対象外となるファイルの扱いは、動作環境ファイルの「monitorexcept」パラメタの指定に従います。</p>

なお、複数のイベントを監視し、処理プロセスの実行対象となるファイルがすべて到着するまで待ち合わせを行う場合は、イベント監視対象ごとにデータ収集定義を準備してください。1つのデータ収集定義につき、1データソースのみ指定できます。

各項目の詳細は、“III Studio ヘルプ”を参照してください。

また、イベント監視を行う場合のプロセス定義の設定内容については、“6.4 処理プロセスの処理手順の設計”を参照してください。

#### 「データボックスの格納方法」に「集約型」を指定し、同時に複数のイベントが発生する場合

対応する処理プロセスをイベントの数だけ同時に実行します。タイミングによって初めに実行した処理プロセスでデータを集約して処理し、他の処理プロセスでは0バイトデータとして扱うことになる場合があります。このような場合は、「0件データの扱い」には「異常」を指定してください。

#### 「0バイトデータの扱い」に「異常」を指定した場合

0バイトデータであるファイルが監視対象ディレクトリに配置され、そのファイルが処理対象ファイルに合致しても対象外ファイルとして処理します。

## データソース定義の参照設定

データソース定義の参照設定は、以下のとおりです。

1. データ収集定義の[プロパティビュー]で、[収集定義]-[データソース一覧]にある[追加]をクリックします。  
→参照するデータソース定義の定義IDの指定および収集元システムのタイプを選択する画面が表示されます。

### ポイント

#### 収集元のタイプ

収集元システムのデータ形式にあわせて設定します。設定値は、以下のとおりです。

- データベース
- ファイル転送
- Salesforce.com
- Interstage Information Storage
- Azure Storage(BLOB)
- Interstage Service Integrator

2. 定義IDとタイプを入力します。  
→[完了]をクリックすると、データソース定義の参照設定が行われます。

## 収集データの設定

以下の定義IDをそれぞれデータ収集定義に記述します。

- 収集データの格納先となる「データボックス」
- 収集データのデータ構造定義の「データ構造定義ID」
- 収集データのフォーマット定義の「フォーマットID」

### ポイント

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- ファイル形式またはサービスバス連携形式のデータ収集の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合は、「フォーマットID」の指定は不要です。

- オブジェクト抽出によるデータ収集の場合
- BLOB収集によるデータ収集の場合

## 収集処理に必要なパラメタの設計

データ収集定義の[プロパティビュー]に表示された[収集定義]-[データソース一覧]から、必要なパラメタを設定するデータソース定義IDを選択します。[データソース詳細]が表示されます。収集処理に必要なパラメタをそれぞれ設定します。

### 注意

#### Excelファイルを収集する場合

Excelファイルを収集する場合、Excelファイルは、上記の手順でデータボックスへの収集を行います。後続ファンクションでデータ変換やDBなどの連携を行うには、上記の手順での収集処理のあとに、プラグインによるExcelデータからCSVデータへの変換処理が必要です。



Excelファイルで利用するプラグインについての詳細は、“[付録K プラグイン機能を利用したExcel連携](#)”を参照してください。

### サービスバス連携入力を行う場合

サービスバス連携入力を行う場合、Service IntegratorのISI運用管理コンソールの表示において、「特定のシーケンス実行に対応する処理プロセスを監視する場合」のInformation Integratorの処理プロセスの見え方がデータ収集定義の「データボックス格納方法」の指定値によって異なることがあります。

異常終了した特定のシーケンス実行に対応する処理プロセスをリカバリする前に、後続の同一シーケンス実行が行われた場合が該当します。

- 「個別型」を指定した場合
  - 特定のシーケンス実行に対応する処理プロセスの履歴には、後続の同一シーケンス実行による処理プロセスの実行結果が表示されます。
  - 後続の同一シーケンス実行に対応する処理プロセスの履歴には、表示すべき実行結果がありません。
- 「集約型」を指定した場合
  - 特定のシーケンス実行に対応する処理プロセスの履歴、および、後続の同一シーケンス実行に対応する処理プロセスの履歴には、後続の同一シーケンス実行による処理プロセスの実行結果が表示されます。

---

## 6.7 データ配付処理の設計

データ配付処理の設計は以下の手順で行います。

1. データの配付先システムの設計
2. 配付データの設計
3. 配付方法の設計

---

### 6.7.1 データ配付先システムの設計

Information Integratorがデータ配付先としてアクセスするシステムの情報(システム名やIPアドレスなど)や、システムへのアクセス手段(データベース情報、ファイル送信のパラメタ、認証情報など)を設計します。

#### データソース定義の作成

[III ナビゲータービュー]でデータソース定義を展開し、設計するデータソース定義IDをダブルクリックします。[エディタ]にデータソース定義の各項目が表示されます。ここで、データソース定義の作成を行います。

#### ポイント

データソースタイプによって、エディタに表示される項目が異なります。データソースタイプは、データソース定義を新規に作成する場合に選択します。詳細は、“[6.3.1 定義の新規作成](#)”を参照してください。

- 配付先システムのデータベースを対象にデータを格納する場合  
配付先システムのデータベースに関するパラメタを入力します。
- 配付先システムにデータをファイル送信で配付する場合  
配付先システムの情報やファイル送信に必要なパラメタを入力します。
- 配付先システムのSalesforceへデータを格納する場合  
収集元システムやオブジェクト格納に必要なパラメタを入力します。
- 配付先システムのInterstage List Creatorで帳票出力を行う場合  
配付先システムや帳票出力に必要なパラメタを入力します。

- 配付先システムのInformation Storageへデータを格納する場合  
配付先システムやIIS格納に必要なパラメタを入力します。
- 配付先システムのAzure Storageへデータを格納する場合  
配付先システムやBLOB配付に必要なパラメタを入力します。
- 配付先システムのService Integratorへデータを配付する場合  
配付先システムやService Integrator配付に必要なパラメタを入力します。

## ポイント

### Service Integratorへデータを配付する場合

サービスバス連携出力を行う場合は、Service Integrator側の設定が必要です。Service Integratorの設定についての詳細は、“[付録 J サービスバス連携を行う場合のService Integratorの設定](#)”を参照してください。

前のファンクション処理の出力データに対して、データボックスに格納するだけの処理を行う場合は、データ配付定義の「共通」部分にのみパラメタを入力します。

## ポイント

サーバに登録済みのデータソース定義がある場合、データ配付定義の「データソースID」に、登録済みのデータソース定義の定義IDを指定して、参照利用することが可能です。同じデータソース定義を再度、作成する必要はありません。

## 6.7.2 配付データの設計

### データボックス名の作成

配付するデータが格納されているデータボックス名を指定します。Information Integratorサーバでデータを配付するためには、データボックスが必要です。[エディタ]で作成するデータ配付定義を選択すると、[プロパティビュー]にデータ配付定義の各項目が表示されます。表示された項目のうち、[共通]-[データボックス]にデータボックス名を入力します。

データボックス名は、処理プロセス内で一意の値を設定する必要があります。複数プロセスに同じデータボックス名を設定すると、複数プロセス間でデータを受け渡すことができます。

### 配付データのデータ構造、フォーマットの設計

配付対象のデータに対して、データ構造、フォーマットを設計します。それぞれデータ構造定義、フォーマット定義で設計します。

## ポイント

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- データベース、オブジェクト以外のデータ配付の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、「フォーマットID」の指定は不要です。

- オブジェクト格納によるデータ配付の場合
- BLOB配付によるデータ配付の場合

以下の場合、「データ構造ID」の指定は不要です。

- IIS格納によるデータ格納の場合

## 6.7.3 配付方法の設計

---

配付方法の設計は、データ配付定義を利用して行います。以下に設計手順を説明します。

### 配付定義の作成

[III ナビゲータービュー]で処理プロセスを展開し、設計する処理プロセス定義IDをダブルクリックします。[エディタ]に処理プロセスの処理手順が表示されます。[エディタ]で作成するデータ配付定義を選択すると、[プロパティビュー]に配付定義の各項目が表示されます。ここで、データ配付定義を作成します。

### データソース定義の参照設定

データソース定義の参照設定は、以下のとおりです。

1. データ配付定義の[プロパティビュー]で、[配付定義]-[データソース一覧]にある[追加]をクリックします。  
→参照するデータソース定義の定義IDの指定および配付先システムのタイプを選択する画面が表示されます。

#### ポイント

---

##### 配付先のタイプ

配付先システムのデータ形式にあわせて設定します。設定値は、以下のとおりです。

- データベース
  - ファイル転送
  - Salesforce.com
  - Interstage Information Storage
  - Interstage List Creator
  - Azure Storage(BLOB)
  - Interstage Service Integrator
- 

2. 定義IDとタイプを入力します。  
→[完了]をクリックすると、データソース定義の参照設定が行われます。

### 配付データの設定

以下の定義IDをそれぞれデータ配付定義に記述します。

- 配付データが格納されている「データボックス」
- 配付データのデータ構造定義の「データ構造定義ID」
- 配付データのフォーマット定義の「フォーマットID」

#### ポイント

---

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- データベース、オブジェクト以外のデータ配付の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、「フォーマットID」の指定は不要です。

- オブジェクト格納によるデータ配付の場合
- BLOB配付によるデータ配付の場合

以下の場合、「データ構造ID」の指定は不要です。

- ・ IIS格納によるデータ格納の場合

データ配信定義の[プロパティビュー]に表示された[配信定義]-[データソース一覧]から、必要なパラメタを設定するデータソース定義IDを選択します。[データソース詳細]が表示されます。配信処理に必要なパラメタをそれぞれ設定します。

## 注意

### Excelファイルを配信する場合

Excelファイルを配信する場合、Excelファイルは、上記の手順でデータボックスへの配信を行います。Excelファイルを配信するには、上記の手順での配信処理の前に、プラグインによるCSVデータからExcelデータへの変換処理が必要です。

Excelファイルで利用するプラグインについての詳細は、“[付録K プラグイン機能を利用したExcel連携](#)”を参照してください。

## 6.8 処理データの構造・フォーマットの設計

処理データの構造、フォーマットの設計手順について説明します。

### 6.8.1 処理データの設計概要

処理するデータに対して、データ構造およびフォーマットを設計します。

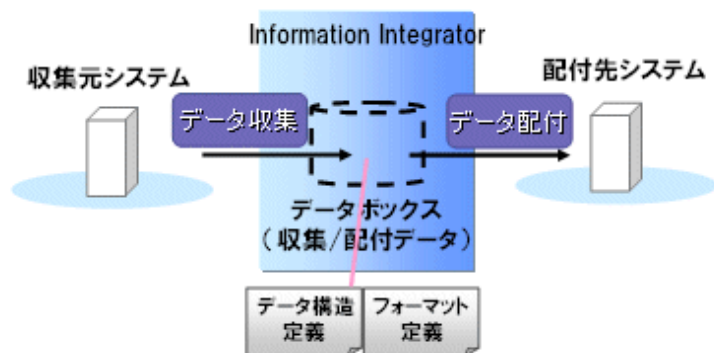
データ構造およびフォーマットの設計が必要となる対象のデータは、処理プロセスの内容によって異なります。処理プロセスのパターンごとに以下に説明します。

#### データ変換処理を含まない処理プロセスの場合

収集データをそのまま配信する場合は、処理データの設計は不要です。

データベースから収集する場合、または収集データに対してデータ構造やフォーマットを変更したデータを配信する場合は、配信するデータの構造、フォーマットを設計します。

図6.3 データ変換処理を含まない場合の設計イメージ

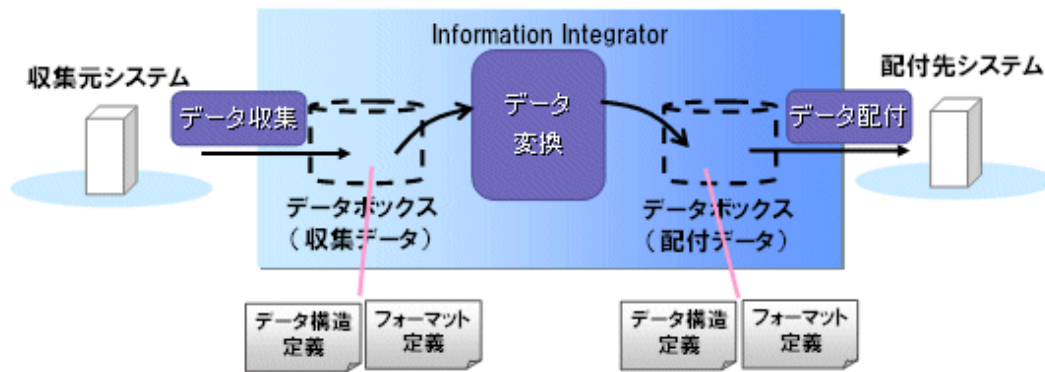


#### 1つだけのデータ変換処理を含む処理プロセスの場合

収集データと配信データのデータ構造およびフォーマットを設計します。

収集データがデータ変換処理の変換前データ、配信データがデータ変換処理の変換後データとなります。

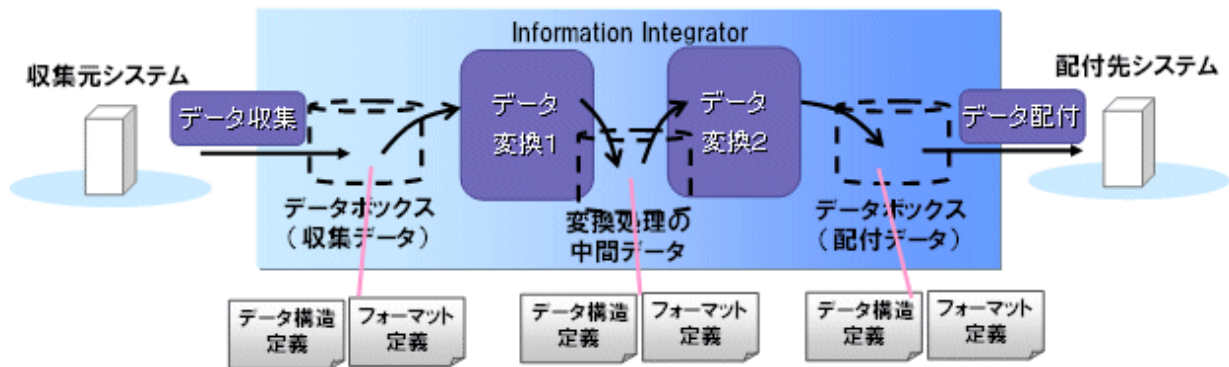
図6.4 1つだけのデータ変換処理を含む場合の設計イメージ



**複数のデータ変換処理(およびデータ振分処理)を含む処理プロセスの場合**

収集データと配信データの他に、データ変換処理の中間結果データに対しても設計を行います。

図6.5 複数のデータ変換処理(およびデータ振分処理)を含む場合の設計イメージ



**6.8.2 操作手順**

以下に操作手順を説明します。

**6.8.3 データ構造の設計**

データ構造の設計では、データ構造定義を利用し、Information Integratorのデータ変換処理で扱うデータに対して項目の並び順やデータ属性を設計していきます。

1. データ構造定義の作成

[III ナビゲータービュー]でデータ構造定義を展開し、設計するデータ構造定義IDをダブルクリックします。[エディタ]のタブに定義ID名が表示され、データ構造定義の各項目が表示されます。ここで、データ構造定義の作成を行います。

## 2. データ構造の設計

[エディタ]に表示されたデータ構造定義の各項目情報(データの項目名、データ属性、桁数など)を入力します。

項番	項目名*	説明	データ型*	全体桁*	小数桁	符号有無	符号位置	符号表現	null制約*	一意性*	
											追加
											削除
											挿入
											上へ

メタ収集機能を利用して相手側システムの情報を取り込むときの手順については、後述の“メタ収集機能を利用した設計手順”を参照してください。

### オブジェクト格納時の留意点

オブジェクト格納時の留意点は、それぞれ以下のとおりです。

- データ追加を行うとき

レコード管理IDは、データ追加時にSalesforceで自動採番されます。

そのため、データ構造定義の項目名およびデータに、Salesforceのレコード管理IDを付与する必要はありません。データ構造定義の項目名にレコード管理IDを含めて定義した場合、データ構造定義に定義したレコード管理IDは無効となります。

- データ更新を行うとき

データ構造定義の項目名およびデータに、Salesforceのレコード管理IDを付与してください。

付与したレコード管理IDとSalesforce側のレコード管理IDが一致するレコードのデータが更新されます。レコード管理IDが一致しない場合は、データは更新されません。異常メッセージ出力後、処理は異常終了します。

- データ追加/更新を行うとき

データ構造定義の項目名およびデータに、Salesforceの外部IDを付与してください。

付与した外部IDとSalesforce側の外部IDが一致するレコードのデータが更新されます。外部IDが一致しないレコードは、データが追加されます。



#### レコード管理IDの付与について

データ追加/更新を行うときは、レコード管理IDの定義は不要です。データ構造定義の項目名にレコード管理IDを含めて定義した場合、データ構造定義に定義したレコード管理IDは無効となります。

## 6.8.4 メタ収集機能を利用した設計手順

メタ収集機能を利用して相手側システムの情報を取り込むときの手順について説明します。



## CSVファイル

1. [エディタ]に表示されたデータ構造定義の各項目情報にある [外部メタ読み込み]-[CSVファイル]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(CSV)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、III Studioの各項目に設定されたツールチップ、および“III Studio ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)が[エディタ]に表示されます。表示内容を確認してください。

## COPY LIBファイル

COBOLのデータ項目に関する定義(COBOL COPY句)が記述されたファイルのメタ収集定義の手順は、以下のとおりです。

1. [エディタ]に表示されたデータ構造定義の各項目情報にある [外部メタ読み込み]-[CSVファイル]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(COBOL COPY句)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、III Studioの各項目に設定されたツールチップ、および“III Studio ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)が[エディタ]に表示されます。表示内容を確認してください。

## データベース

1. [エディタ]に表示されたデータ構造定義の各項目情報にある [外部メタ読み込み]-[データベース]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(データベース)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、III Studioの各項目に設定されたツールチップ、および“III Studio ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)が[エディタ]に表示されます。

表示された内容は、Information Integratorサーバで利用可能な属性になっているかなどを確認してください。Information Integratorサーバで利用可能な属性については、“[B.1.4 Information Integratorで収集できるデータベース形式](#)”または“[B.2.4 Information Integratorで配付できるデータベース形式](#)”を参照してください。

## オブジェクト

1. [エディタ]に表示されたデータ構造定義の各項目情報にある [外部メタ読み込み]-[salesforce.com]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(salesforce.com)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、III Studioの各項目に設定されたツールチップ、および“III Studio ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)が[エディタ]に表示されます。表示内容を確認してください。
3. オブジェクト格納で利用するメタの収集では、オブジェクト格納のアクセス権の有無に関わらずデータ構造定義に情報を表示します。アクセス権がない項目は、オブジェクト格納を行うことができないため、アクセス権のない項目は削除してください。アクセス権のない項目が記載された状態でオブジェクト格納を行うとエラーとなります。

アクセス権についての設定は、“[9.4 オブジェクト抽出・格納利用時のセキュリティ設定情報](#)”を参照してください。

なお、以下の項目は、オブジェクト格納のアクセス権はありません。必ず削除してください。

- IsDeleted
- CreatedDate
- LastModifiedDate
- SystemModstamp

## 帳票

1. [エディタ]に表示されたデータ構造定義の各項目情報にある [外部メタ読み込み]-[List Creator帳票定義体]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(List Creator帳票定義体)]画面が表示されます。

2. メタの収集元(Interstage List Creatorの帳票定義体)を入力します。

## ポイント

### 帳票定義体について

- 帳票定義体(拡張子.pmd)は、一般的に以下に格納されています。

Interstage List Creator インストールディレクトリ¥Assets

- 帳票定義体で「集団項目を使用」が指定されている場合、III Studioの項目には、集団項目を表示します。帳票定義体で「構成項目を使用」が指定されている場合、III Studioの項目には、構成項目を表示します。

→収集したメタ情報(項目名や属性情報)が[エディタ]に表示されます。表示内容を確認してください。

## ポイント

- 演算結果や定数を変換処理後のデータの項目として出力する場合には、変換後のデータに対してデータ構造を設計するときに、出力項目を含めたデータ構造として設計します。
- 同じようなデータ構造定義を作成する場合には、作成したデータ構造定義を複製し、「データ構造定義ID」を変更して流用することにより、簡単に作成することができます。

## 6.8.5 データフォーマットの設計

データフォーマットの設計では、フォーマット定義を利用し、データの形式や文字コードなどの属性を定義します。

### 1. フォーマット定義の作成

[III ナビゲータービュー]でフォーマット定義を展開し、設計するフォーマット定義IDをダブルクリックします。[エディタ]のタブに定義ID名が表示され、フォーマット定義の各項目が表示されます。ここで、フォーマット定義の作成を行います。

### 2. データフォーマットの設計

[エディタ]に表示されたフォーマット定義の各項目情報(文字コード、レコードタイプ、区切り文字など)を入力します。

## 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、留意事項があります。留意事項については、「[6.5.3 見出し行の存在するデータを扱う場合の留意事項](#)」を参照してください。

## 6.9 データ変換処理の設計

データ変換処理の設計手順を説明します。

設計する変換処理には、以下の2種類があります。

### • データ変換処理の設計

1つ、または複数の入力データに対して、項目編集やレコード編集を行い、1つのデータに変換する処理を設計します。データ変換定義を利用して設計を行います。

データ変換処理の設計は、以下の手順で行います。

#### — 変換前後のデータ構造およびフォーマットの設計

変換前と変換後のそれぞれのデータ構造やフォーマットを設計することにより変換処理を設計します。

データ構造やフォーマットは、それぞれデータ構造定義、フォーマット定義で設計を行い、データ変換定義で各定義IDを指定することで設計を行います。



変換前後のデータ構造やフォーマットの差異に対して、Information Integratorが必要な変換処理を行います。例えば、変換前のフォーマット定義の「文字コード」に「SJMS」、変換後のフォーマット定義の「文字コード」に「UTF8」と定義することによって、Shift-JISからUTF8への変換処理を設計できます。

#### 一 項目編集、レコード編集処理の設計

演算式、結合、ソート、または集計などの変換処理を、データ変換定義の各フィールドにパラメタ入力することで設計します。1つのデータ変換定義上で、複数のデータ変換処理を組み合わせることで定義することができます。

##### ・ データ振分処理

1つのデータをレコード条件によって複数のデータに分割する処理、同じデータを複数のシステムに配付する処理を設計します。データ振分定義を利用して設計を行います。

## ポイント

### データ変換の処理順序について

ファイル全体のフォーマットに影響するフォーマット変換、文字コード変換は、データ変換の最初に実行します。その次に、項目編集、レコード編集の処理を実行します。集計、ソートはデータ変換の最後に実行します。

一度にすべての変換処理が定義できない場合や、複数の収集データをそれぞれ別の変換処理をした後に、1つのデータに変換する場合などは、データ変換定義を複数作成します(複数の変換処理を行う)。

処理プロセスの処理手順設計時に、作成した複数の変換ファンクションの処理順序を決めます。

## ポイント

### 複数のデータ変換定義が必要な変換のパターン

以下のような変換処理を設計する場合は、複数のデータ変換定義が必要になります。

- ・ ソートや集計の後に項目編集、連結、レコード抽出を実行する場合

## 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、留意事項があります。留意事項については、“[6.5.3 見出し行の存在するデータを扱う場合の留意事項](#)”を参照してください。

## 6.9.1 データ変換処理の設計

変換前後のデータに対してデータ構造とフォーマットの設計が完了したら、データ変換定義を作成し、データ変換処理の設計を行います。

データ変換処理の設計は、あらかじめ設計したデータ構造定義を読み込んで行うため、データ構造定義を利用して変換前と、変換後のデータに対するデータ構造の設計を完了しておく必要があります。

### 6.9.1.1 データ変換定義の作成

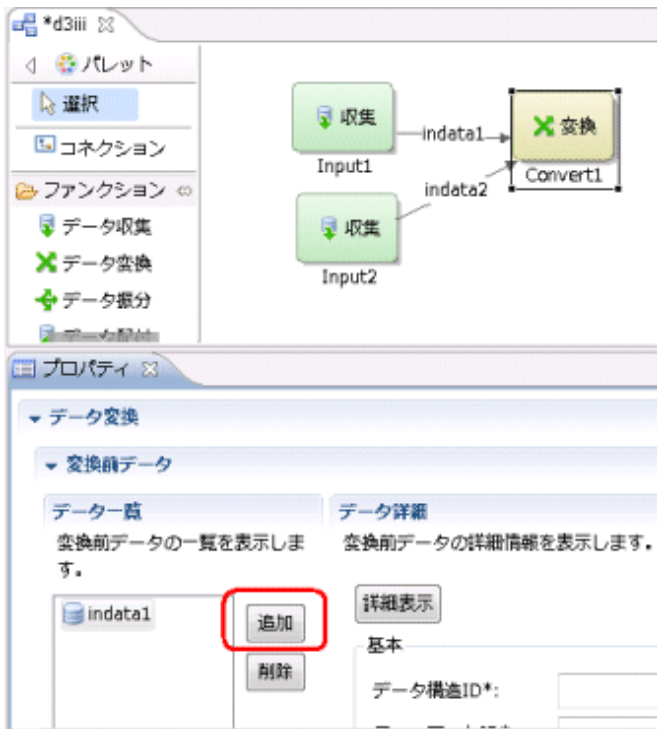
[III ナビゲータービュー]で処理プロセス定義を展開し、設計する処理プロセス定義IDをダブルクリックします。[エディタ]に処理プロセスの処理手順が表示されます。[エディタ]で作成するデータ変換定義を選択すると、[プロパティビュー]に変換定義の各項目が表示されます。ここで、データ変換定義を作成します。

### 6.9.1.2 データ構造の読み込み

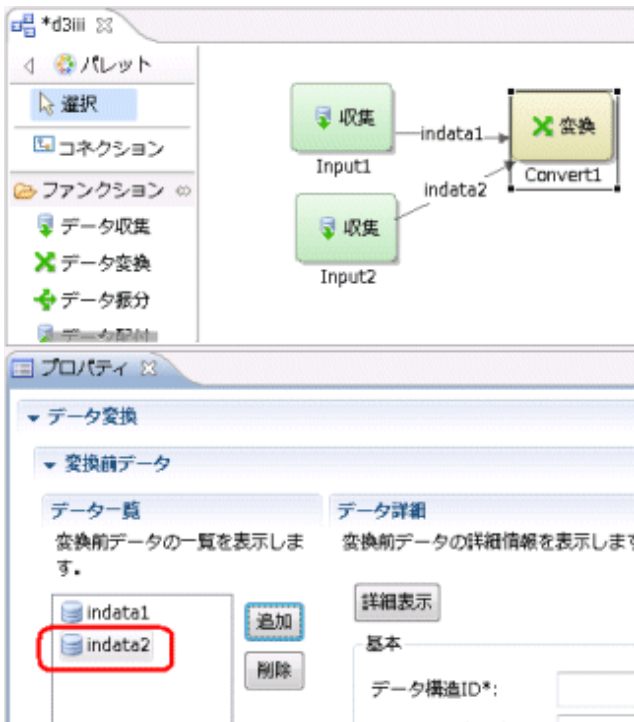
#### データ構造の作成

変換前のデータ(変換処理の入力となるデータ)と変換後のデータ(変換処理の出力となるデータ)に対して、それぞれデータ構造を作成します。変換前のデータが2つ以上の場合は、以下の手順でデータ構造を追加します。

1. [データ変換定義]-[変換前データ]-[データ一覧]で、[追加]ボタンをクリックします。



2. [データ一覧]にデータ構造が1つ追加されます。



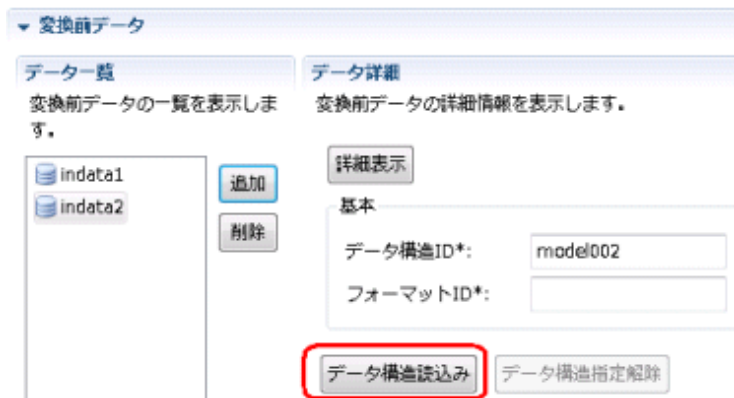
3. 1、2の操作を繰り返し、入力データ数分のデータ構造を追加します。

## データ構造定義の読み込み

作成した入力データに対し、あらかじめ作成したデータ構造定義を、以下の手順で読み込みます。

1. [データ変換定義]-[変換前データ]-[データ一覧]から、設定を行いたい入力データを選択します。  
→[データ詳細]に、入力データの詳細指定画面が表示されます。

2. [データ詳細]の[データ構造ID]に、読み込むデータ構造定義の「データ構造ID」を指定します。
3. [データ構造読み込み]ボタンをクリックします。



- 指定した[データ構造ID]に該当するデータ構造定義が、データ変換定義と同じプロジェクト上にある場合は、データ構造定義の内容から、以下を読み込んで表示します。
  - 「項目名」
  - 「データ型」
  - 「全体桁」
  - 「小数桁」
- 指定した「データ構造定義ID」に該当するデータ構造定義が、データ変換定義と同じプロジェクト上にない場合は、以下のメッセージが表示されます。

指定したデータ構造定義が対象のプロジェクトに存在しません。サーバからデータ構造を読み込みますか？

[OK]ボタンをクリックすると、Information Integratorサーバに登録したデータ構造定義を読み込むことができます。

(サーバへ接続していない場合は、[サーバ接続]ダイアログが表示されるので、Information Integratorサーバへ接続してください)

同様の操作を出力データ(変換後データ)に対しても実施します。

### 注意

- データ構造を読み込んだフィールドに対して編集操作できません。データ構造を変更する場合は、データ構造定義を修正し、Information Integratorサーバに登録した後に、再度データ構造を読み込んでください。
- データ構造を再び読み込むと、項目の対応付け(マッピング情報)や演算式情報が削除されるので、再度設定してください。

## 6.9.1.3 データフォーマットの設定

変換前のデータ、変換後のデータそれぞれに対してフォーマットを定義します。[データ詳細]の「フォーマットID」に、あらかじめ設計したフォーマット定義の「フォーマットID」を指定します。

## 6.9.1.4 項目対応付けの設計手順

変換前のデータと変換後のデータに対して、各項目の対応付け(マッピング)を行います。

データ構造を読み込むと、変換前、変換後データの各項目に対して自動で項番が設定されます。

変換後データの設定エリアの「項番(from)」に変換前データの項番を入力し、「項番(from)」と「項番(to)」を並べて設定することにより、変換前後の項目の対応付けを定義します。

以下の対応付けを行う場合の、変換後データの「項番(from)」の入力例を示します。

- ・ 変換前データの1番目から3番目のデータを、逆順で変換後データの1番目から3番目に出力

項番(from)	項番(to)	ラベル	項目名
1-3	1		機種名
1-2	2		店舗名
1-1	3		仕入...

### 6.9.1.5 項目編集処理の設計手順

各項目に対して演算式を指定します。

編集対象の項目の演算式フィールドに、演算種別、引数を指定します。

#### ポイント

データ構造定義のデータ形式が「一般」であり、変換後データの設定で演算式の演算種別に「四則演算」、「文字列の連結」を利用する場合、変換前データの項目を使った演算が可能です。変換前データの項目を使った演算は、ラベル(@n)を利用します。

#### 【設定例】

粗利益: 小数点以下1桁で切り捨てます。

店舗コード: データの左側の、連続する文字「0」を取り除きます

店舗名: 半角文字を全角文字に変換します。

[変換前のデータの設定]

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	演算種別	引数
1-1		粗利益		SQL_...	10	3	切り捨て	1
1-2		店舗...		SQL_...	12		トリム	左0
1-3		店舗名		SQL_...	12		全角/半角変換	全角文字

演算式に利用する項目には、ラベル(@n: nは1から1000までの昇順の数字)を指定し、演算式の引数にラベルを指定します。また、復帰情報番号(¥n: nは1から1000までの昇順の数字)を指定することで、入力データを編集せずに、演算結果を出力データに指定することができます。

#### 【設定例】

売上高と個数のデータに対する四則演算を行います。

売上高のデータに対する編集はせずに、四則演算を行い、結果を出力します。

[変換前のデータの設定]

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	演算種別	引数	演算結果
1-1	@1	売上高		SQL_...	10		四則演算	(@1+100)/@2	¥1
1-2	@2	個数		SQL_...	12				

変換後の出力データの項目の「項目(from)」に演算結果変数「¥1」を指定します。

#### 【設定例】

四則演算の結果を、変換後データの4番目の項目「単価」として出力します。

[変換後データの設定]

項番(from)	項番(to)	ラベル	項目名
1-1	1		売上高
1-2	2		個数
¥1	3		単価

項目に対して複数の演算を行う場合には、演算式の入力フィールドで、右クリックメニューから[演算式追加]を選択して、演算式の入力フィールドを追加します。

演算結果変数を利用することで、入力データに対して複数の演算結果を出力することが可能です。

#### 【設定例】

仕入先コードの1～3文字と、4～6文字を、それぞれ別項目に出力します。

[変換前データの設定]

項番	ラベル	項目名	演算種別	引数	演算結果変数	演算種別	引数	演算結果変数
1-1		売上高						
1-2		仕入先コード	文字列の部分抽出	1-3	¥1	文字列...	4-6	¥2

[変換後データの設定]

項番(from)	項番(to)	ラベル	項目名
1-1	1		売上高
¥1	2		仕入先コード1
¥2	3		仕入先コード2

#### 【設定例】

名字と名前を連結して、フルネームで出力します。

#### 変換前データの設定

ラベル	項目名	説明	データ型	全体桁
@1	名字		SQL_...	10
@2	名前		SQL_...	10

#### 変換後データの設定

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	演算種別	引数
1		フルネーム		SQL_CHAR	20		文字列の連結	@1+@2

#### 【設定例】

販売数と単価から売り上げを計算して出力します。

#### 変換前データの設定

項番	ラベル	項目名	説明	データ型	全体桁	小数
1-1		品名		SQL_CHAR	10	
1-2	@1	販売数		SQL_INTEGER		

項番	ラベル	項目名	説明	データ型	全体桁	小数
2-1		品名		SQL_...	20	
2-2	@2	単価		SQL_I...		

#### 変換後データの設定

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	演算種別	引数
1		品名		SQL_...	10			
2		売り...		SQL_I...			四則演算	@1*@2

#### 結合指定

##### ▼ 結合指定

結合型: INNER JOIN ▼

項番	キー1	キー2	結合条件
1	1-1	2-1	完全一致



#### 注意

横連結(INNER JOIN/OUTER JOIN)では、変換前データの各項目を使って演算できます。縦連結(UNION/UNION ALL)では指定できません。

### 6.9.1.6 ソート処理の設計手順

変換後のデータとして出力する項目に対して、ソートキーとなる項目を指定します。

ソートの種別(昇順/降順)と、ソートキーの優先順位を組み合わせで設定します。

#### 【設定例】

以下の例は、変換後のデータに対して、「1:店舗名を昇順」、「2:機種名を降順」、「3:配送先を昇順」の優先順位で並べ替えて出力する指定です。

項番	ラベル	項目名	演算種別	引数	演算種別	引数	演算種別	引数	演算種別	引数	集計	ソート
1		機種名										2:昇順
2		店舗名										1:昇順
3		仕入先名										3:昇順

### 6.9.1.7 集計処理の設計手順

変換後のデータとして出力項目に対して、集計対象とする項目および、集計時にグルーピングする項目を指定します。グルーピングした項目単位で合計を「出力する/しない」を指定できます。

#### 【設定例】

以下の例は、変換後データを、店舗名および機種名でグルーピングを行い、売上高の店舗名ごとの小計、および全体合計を出力する指定です。

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	集計
1		機種名		SQL_...	10	3	2
2		店舗名		SQL_...	50		1:小計あり
3		売上高		SQL_...	25		グループ計

### 6.9.1.8 変換処理の詳細設定の設計手順

データ変換処理に関する詳細設定について説明します。

データ変換処理実行時の詳細な動作オプションとして、以下のような設定ができます。

#### 【詳細設定の例】

- ・ 変換処理エラー時の動作(処理継続、処理中断)
- ・ 文字コード変換エラー時の代替文字

変換動作定義の設定フィールドにパラメタを入力して設定します。

### 6.9.1.9 結合処理の設計手順

以下の手順で、結合指定のフィールドに、結合型、結合キー、結合条件をそれぞれ指定します。

1. 「結合型」に、結合処理の種類を指定します。
2. 結合処理の種類が「UNION ALL」以外の場合は、結合指定行を追加します。  
「結合型」直下のテーブルの[追加]ボタンをクリックします

▼ 結合指定

結合型: INNER JOIN ▼

項番	キー-1	キー-2	結合条件

追加

3. 追加された結合指定行に対して、結合キー、結合条件を指定します。

#### ー INNER JOIN/OUTER JOINの場合

結合する2つの入力データに対して、それぞれ結合キーを指定します。

複数の結合を行う場合は、結合指定行をさらに追加して指定します。

#### 【設定例】

以下の結合を行う例を示します。

- 1番目の入力データの2番目の項目と、2番目の入力データの3番目の項目を結合、かつ
- 1番目の入力データの3番目の項目と、3番目の入力データの5番目の項目を結合

▼ 結合指定

結合型: INNER JOIN ▼

項番	キー-1	キー-2	結合条件
1	1-2	2-3	完全一致
2	1-2	3-5	完全一致



— UNIONの場合

重複行を判定する際に対象とする項目を「結合キー」に指定します。

結合キーには、1番目の入力データ(indata1)の項目を指定してください。(例: 1-1,1-2)

結合キーは、最大8個まで指定できます。ただし、入力データの項目数が8個以下の場合は、項目数-1個まで指定できます。

— UNION ALLの場合

結合キーや結合条件の指定は不要のため、結合指定行は指定しません。

## 6.9.2 データ振分処理の設計

---

レコード条件で複数のデータに分割するデータ処理は、データ振分処理として設計します。データ振分処理は、データ振分定義で設計します。

### 6.9.2.1 データ振分定義の作成

[III ナビゲータービュー]で処理プロセスを展開し、設計する処理プロセス定義IDをダブルクリックします。[エディタ]に処理プロセスの処理手順が表示されます。[エディタ]で作成するデータ振分定義を選択すると、[プロパティビュー]にデータ振分定義の各項目が表示されます。ここで、データ振分定義を作成します。

### 6.9.2.2 データ振分処理の設計

[プロパティビュー]の[振分設定]に、振分条件が必要な分だけ追加して、振分条件を指定します。

振分後のデータに対する「振分データID」が自動設定されるので、処理プロセスの処理順序設計時にデータ振分処理の前後のファンクション間でのデータ受渡しの対応関係を設計します。

#### ポイント

- 「振分条件」に「other」を指定することができます。「振分条件」に「other」を指定した場合、前述に指定されているすべての振分条件に属さないデータが振分されます。
- 「other」を含めた振分条件の数は、5以内です。

#### 注意

- 複数のデータ振分処理を先行処理として選択することはできません。
- 「振分条件」に「other」を指定する場合は、以下を注意してください。
  - 「other」を指定する場合には、1つ以上の振分条件が設定されている必要があります。
  - 「other」は、「振分条件」の最後に指定してください。

## 6.10 プラグイン処理の設計

---

III Studioを利用したプラグイン処理の設計手順について説明します。

### 6.10.1 処理プラグイン機能の設計手順

---

プラグイン処理(外部プログラムの呼び出し)の設計は、以下の手順で行います。

1. プラグイン定義の作成

[III ナビゲータービュー]の[プラグイン定義]を展開して、作成するプラグイン定義ID名をダブルクリックします。[エディタ]のタブに定義ID名が表示され、プラグイン定義の各項目が表示されます。

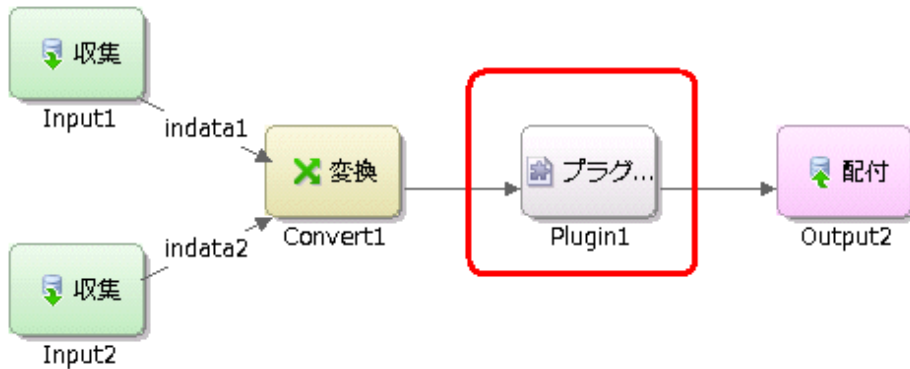


## P ポイント

プラグイン種別によって、エディタに表示される項目が異なります。プラグイン種別は、プラグイン定義を新規に作成する場合に選択します。詳細は、“6.3.1 定義の新規作成”を参照してください。

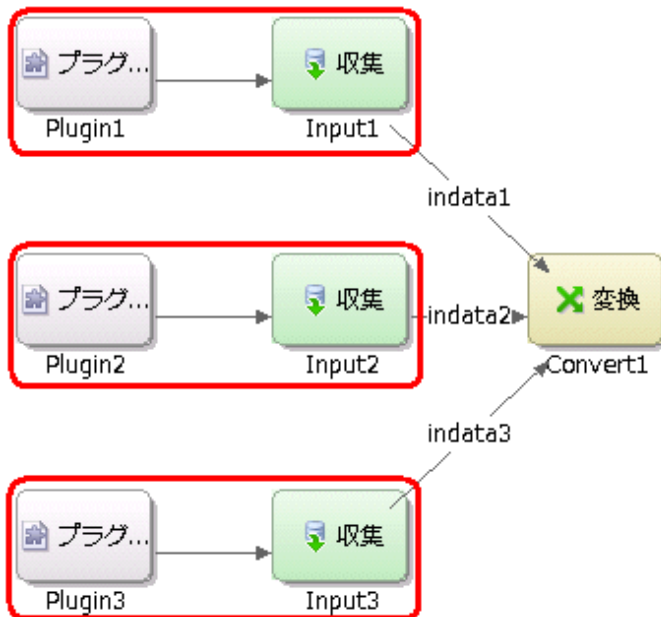
### 2. プラグイン処理を処理プロセスの手順に追加

処理プロセスの処理順序の設計時に、プラグイン処理を追加します。



## 注意

プラグイン処理を先頭の処理に指定する場合、後続処理に、対応するデータ収集処理を必ず指定してください。また、その場合、後続のデータ収集処理が複数存在する場合は、プラグイン処理を、データ収集処理の数分用意してください。



## 6.10.2 変換プラグイン機能の設計手順

プラグイン処理(外部プログラムの呼び出し)の設計は、以下の手順で行います。

### 1. プラグイン定義の作成

[III ナビゲータービュー]の[プラグイン定義]を展開して、作成するプラグイン定義ID名をダブルクリックします。[エディタ]のタブに定義ID名が表示され、プラグイン定義の各項目が表示されます。

## ポイント

プラグイン種別によって、エディタに表示される項目が異なります。プラグイン種別は、プラグイン定義を新規に作成する場合に選択します。詳細は、“[6.3.1 定義の新規作成](#)”を参照してください。

### 2. プラグイン処理を変換定義の処理に追加

変換定義の項目別の処理手順に、プラグイン処理を追加します。

項番	ラベル	項目名	説明	データ型	全体桁	小数桁	演算種別	引数
1-1		機種名		SQL_...	10	3		
1-2		店舗名		SQL_...	50		プラグイン	plg01(\$店...
1-3		売上高		SQL_...	25			

各項目の入力内容については、“III Studio ヘルプ”を参照してください。

## 6.10.3 データを圧縮または解凍する場合

相手側システムから圧縮されたファイルまたはBLOBデータを収集したとき、収集したデータを解凍します。また、相手側システムへファイルまたはBLOBデータを配付するとき、配付するデータを圧縮します。

データ圧縮およびデータ解凍のプラグインの設計手順は、処理プラグイン機能の設計手順と同じです。詳細は、“[6.10.1 処理プラグイン機能の設計手順](#)”を参照してください。

データ圧縮およびデータ解凍を行う場合、プラグイン定義の作成時に、データ圧縮コマンドまたはデータ解凍コマンドを指定します。コマンドについて説明します。

プラグイン定義へのコマンドの入力方法は、“III Studio ヘルプ”を参照してください。

### 6.10.3.1 ificompcmd(データ圧縮コマンド)

#### 説明

zip形式でデータを圧縮します。

#### コマンドの格納先

- Windowsの場合

「ificompcmd.cmd」は、以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥bin
```

- UNIXの場合

「ificompcmd.sh」は、以下のディレクトリにあります。

```
/opt/FJSVifisv/bin/
```

#### コマンドの形式

- Windowsの場合

```
ificompcmd.cmd 入力ファイル名 出力ファイル名
```

- UNIXの場合

```
Ificompcmd.sh 入力ファイル名 出力ファイル名
```

## オペランドの説明

### 入力ファイル名

入力データ(圧縮前のデータ)のファイル名を255バイト以内の絶対パスで指定します。

### 出力ファイル名

出力データ(圧縮後のデータ)のファイル名を255バイト以内の絶対パスで指定します。すでに、ファイルが存在する場合は、上書きします。

## ポイント

### プラグイン定義の指定方法について

- ・ 実行形式プラグインのプラグインパスにコマンドを絶対パスで指定します。入力パラメタ1に「\$IFI\_InputFile」(前処理のファンクションのデータ)、入力パラメタ2に「\$IFI\_OutputFile」(後処理のファンクションのデータ)を指定します。
- ・ 作成したプラグイン定義は、プロセス定義のデータ収集ファンクションとデータ配付ファンクションの間に指定します。

## 復帰値

- 0: 正常終了
- 8: 異常終了
- 12: 異常終了

## 6.10.3.2 ifiuncompcmd(データ解凍コマンド)

### 説明

zip形式で圧縮されたデータを解凍します。

### コマンドの格納先

- ・ Windowsの場合  
「ifiuncompcmd.cmd」は、以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥bin
```

- ・ UNIXの場合  
「ifiuncompcmd.sh」は、以下のディレクトリにあります。

```
/opt/FJSVifisv/bin/
```

### コマンドの形式

- ・ Windowsの場合

```
ifiuncompcmd. cmd 入力ファイル名 出力ファイル名
```

- ・ UNIXの場合

```
Ifiuncompcmd. sh 入力ファイル名 出力ファイル名
```

## オペランドの説明

### 入力ファイル名

入力データ(解凍前のデータ)のファイル名を255バイト以内の絶対パスで指定します。

## 出力ファイル名

出力データ(解凍後のデータ)のファイル名を255バイト以内の絶対パスで指定します。すでに、ファイルが存在する場合は、上書きします。

## ポイント

### プラグイン定義の指定方法について

- ・ 実行形式プラグインのプラグインパスにコマンドを絶対パスで指定します。入力パラメタ1に「\$IFL\_InputFile」(前処理のファンクションのデータ)、入力パラメタ2に「\$IFL\_OutputFile」(後処理のファンクションのデータ)を指定します。
- ・ 作成したプラグイン定義は、プロセス定義のデータ収集ファンクションとデータ配付ファンクションの間に指定します。

## 復帰値

- 0: 正常終了
- 8: 異常終了
- 12: 異常終了

## 6.10.4 ファイル転送エージェントの処理プロセスと連携する場合(オプション機能)

処理プラグイン機能を利用して、各ファイル転送エージェントの処理プロセスと連携します。処理プラグイン機能の設計手順の詳細は、「[6.10.1 処理プラグイン機能の設計手順](#)」を参照してください。処理プラグイン機能を利用して、各ファイル転送エージェントの処理プロセスと連携する場合、連携用のコマンドを指定します。コマンドについて説明します。

プラグイン定義へのコマンドの入力方法は、「III Studio ヘルプ」を参照してください。

### 6.10.4.1 ificlexepc(エージェント処理プロセス連携用コマンド)

#### 説明

ファイル転送エージェントの処理プロセスを実行します。

#### コマンドの格納先

- ・ Windowsの場合  
「ificlexepc.bat」は、以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥bin
```

- ・ UNIXの場合  
「ificlexepc.sh」は、以下のディレクトリにあります。

```
/opt/FJSVifisv/bin/
```

#### コマンドの形式

- ・ Windowsの場合

```
ificlexepc.bat 連携対象のファイル転送エージェントのシステム名 ファイル転送エージェントの処理プロセスID
```

- ・ UNIXの場合

```
ificlexepc.sh 連携対象のファイル転送エージェントのシステム名 ファイル転送エージェントの処理プロセスID
```

## オペランドの説明

### 連携対象のファイル転送エージェントのシステム名

システム名を31文字以内の英数字で指定します。

### ファイル転送エージェントの処理プロセスID

連携先のファイル転送エージェントにある処理プロセスIDを31文字以内の英数字で指定します。

## ポイント

### プラグイン定義の指定方法について

実行形式プラグインのプラグインパスにコマンドを絶対パスで指定します。入力パラメタ1に連携対象のファイル転送エージェントのシステム名、入力パラメタ2にファイル転送エージェントの処理プロセスIDを指定します。

## 復帰値

- 0: 正常終了
- 8: 連携時異常
- 12: 入力パラメタ異常
- 14: 入力パラメタ長異常

## 第7章 デザインシートを利用した設計

デザインシートを利用した処理プロセスの設計手順について説明します。

処理プロセスの定義をデザインシートで行います。デザインシートは、以下の各シートから構成されています。

- 処理プロセスの処理手順を設計する  
プロセス定義シート
- 処理プロセスのパラメタを設計する
  - 処理プロセスに含めるファンクション処理を定義する  
これらのシートは、プロセス定義シートと同じExcelブック内に作成します。
    - データ収集定義シート
    - データ変換定義シート
    - データ振分定義シート
    - データ配付定義シート
  - 上記の各ファンクションで必要となるパラメタを設計する
    - データソース定義シート
    - データ構造定義シート
    - フォーマット定義シート
    - プラグイン定義シート

1つの処理プロセスに対して、1つのExcelブックでデザインシートを作成します。

ただし、ファンクションで必要となるパラメタを設計するデザインシートは、プロセス定義シートとは、別のExcelブックで作成できます。(同じExcelブックに作成してもかまいません。)

デザインシートの利用方法および、各デザインシートの詳細について説明します。

### 7.1 デザインシートの利用方法

デザインシートは、Excelが導入されているPC上に、Information Integratorクライアントをインストールして利用します。

以下に、デザインシートの利用方法を説明します。

なお、デザインシートの各項目についての説明は、“デザインシート操作ヘルプ”を参照してください。

また、デザインシートを利用するために必要なExcelアドインファイルをExcelに登録する必要があります。デザインシート用のExcelアドインファイルの登録方法については、“セットアップガイド”を参照してください。

#### 7.1.1 デザインシートの新規作成

Excelをスタートメニューから起動し、Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]メニューを選択して、デザインシートを作成します。

作成するデザインシートごとのメニューを選択すると、各デザインシートのテンプレートが作成されます。

処理プロセスおよび各ファンクション処理を設計するためのデザインシートは、[定義の新規作成]のサブメニュー[処理プロセス]の中から、作成するデザインシートを選択してデザインシートを作成します。

デザインシートを新規に作成すると、Excel上でアクティブになっているExcelブックの最後に、デザインシートがExcelシートとして追加されます。Excelブックがない場合は、新規にExcelブックを作成した上で、デザインシートが作成されます。

## 注意

Excelを利用して、Information Integratorでの設計作業以外の作業を行っている場合は、編集中のExcelブックをすべて閉じてから、デザインシートを作成してください。

## ポイント

- デザインシートの各定義シートで作成した定義情報は、定義ごとに設定する定義ID(プロセス定義ID、ファンクションIDなど)で管理します。定義IDは、Information Integratorサーバのシステム上で一意である必要があります。そのため、デザインシートで設計を行う際には、各種定義シートに設定する定義IDが重複しないように、システムで一意になる定義IDを、プレフィックスの付け方などの規約により事前に設計しておくことを推奨します。
- 統合管理オプションとファイル転送エージェントを併用する場合は、運用後に処理プロセス間の関連性を確認できるようにIDを付与してください。例えば、ファイル転送エージェントで利用するプロセス定義IDは、連携するプロセス定義ID(Information Integratorサーバで利用する処理プロセス)に通番を付与するなどしてください。

## 7.1.2 デザインシートの入力内容のチェック

デザインシートの各種パラメタを入力後、入力したパラメタをチェックします。

チェックする対象のデザインシートをアクティブにした状態で、Excelアドインメニュー「Integratorデザインシート」の[定義内容チェック]メニューを選択します。

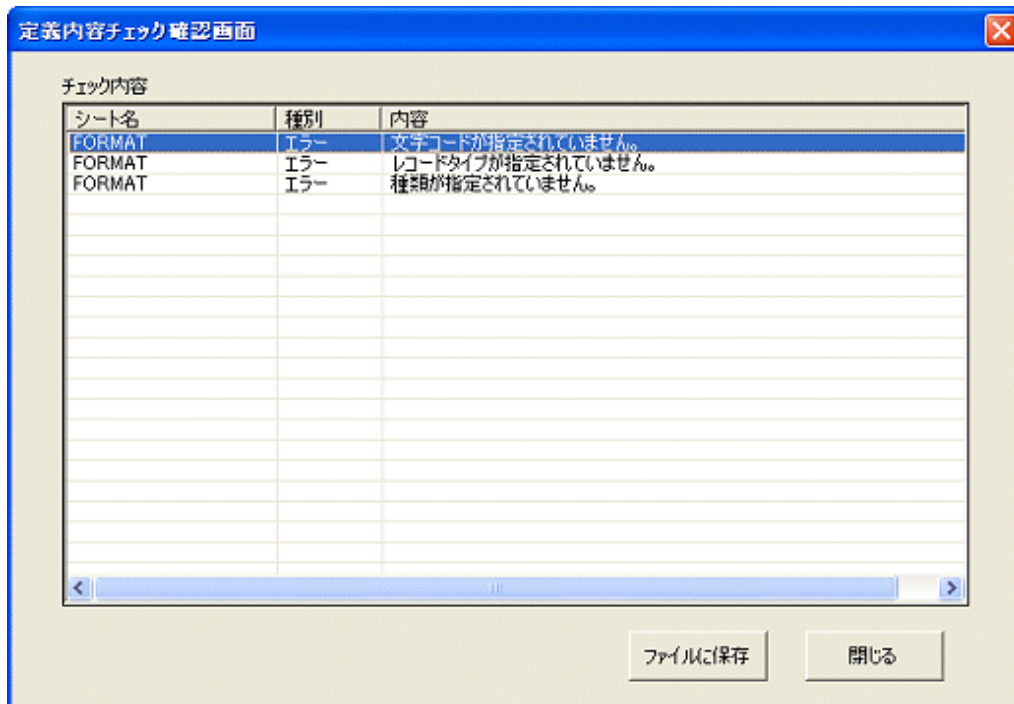
- 入力内容が正しい場合

以下のメッセージが表示されます。

チェックが終了しました

- 入力内容に誤りがある場合

[定義内容チェック確認画面]画面で入力内容の誤りの内容がリスト表示されます。



リストの各行をダブルクリックすると、デザインシート上の該当のセルが選択されるので、入力内容を確認して修正を行います。

## 7.1.3 デザインシートをサーバに登録

デザインシートで設計した内容を、サーバに登録します。

サーバに登録するデザインシートを作成したExcelブックをアクティブにした状態で、Excelアドインメニュー[定義をサーバに登録]メニューを選択します。登録方法の詳細は、“セットアップガイド”を参照してください。

[処理プロセス]を選択すると、Excelブックに作成したプロセス定義シートおよびプロセス定義に指定された各ファンクション処理のデザインシートが一括でサーバに登録されます。その他の定義は、デザインシートごとに登録します。

### ポイント

デザインシートをサーバに登録するには、Information Integratorサーバと接続する必要があります。

また、Information Integratorサーバと接続するには、デザインシートの登録用のセットアップを完了し、Information Integratorサーバを起動する必要があります。デザインシートの登録用のセットアップ、およびInformation Integratorサーバの起動方法については、“セットアップガイド”を参照してください。

デザインシートの登録用セットアップが完了していない場合は、PC上に一時保存してください。デザインシートの一時保存については、“7.1.5 デザインシートの一時保存”を参照してください。

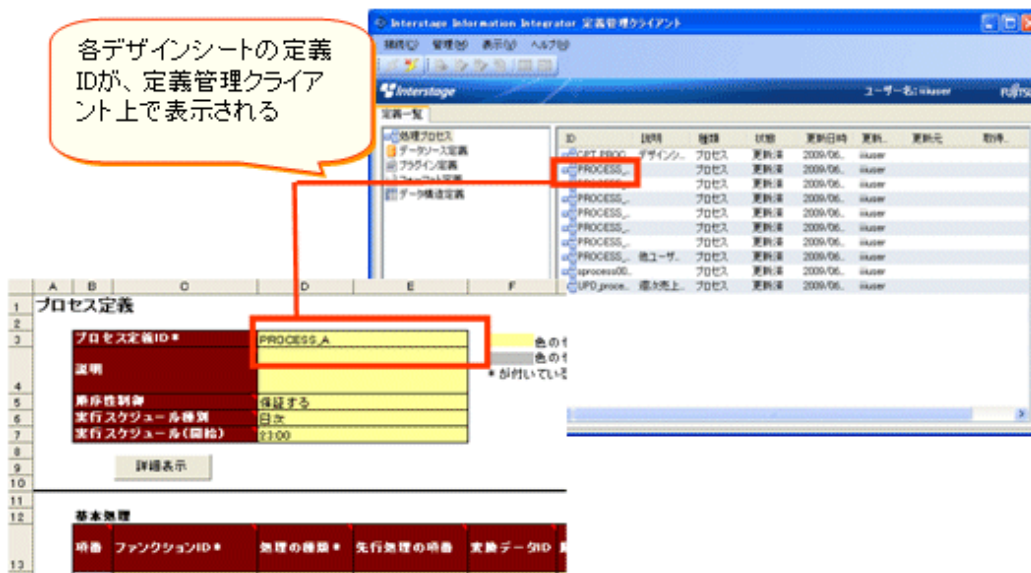
サーバに登録すると、入力内容のチェックを自動的に行います。チェック結果については、“7.1.2 デザインシートの入力内容のチェック”を参照してください。

また、デザインシート間やサーバに登録された定義情報との関係の整合性をチェックし、問題がある場合はエラーメッセージが表示され、エラーの詳細情報がファイルに出力されます。エラーメッセージが表示された場合は、エラーメッセージに記載されたファイルを参照して対処してください。

サーバに登録された定義情報は、定義管理クライアントを利用して一覧で確認することができます。

定義管理クライアント上では、それぞれのデザインシートで入力した定義IDが、表示されます。

図7.1 定義IDと定義管理クライアントの表示の関係



### 注意

サーバに登録した定義は、デザインシートで設定した定義IDで管理されます。

サーバへの登録時に定義IDがなければ新規に登録されます。定義IDがすでに存在する場合は、該当するサーバ上の定義が編集中の状態の場合に、上書きされます。



## 7.1.4 サーバに登録した定義内容の修正方法

デザインシートでInformation Integratorサーバに登録した定義内容を修正する場合は、定義管理クライアントを利用して定義内容をサーバから取得します。定義内容はデザインシートとして取り出すことができるので、デザインシート上で定義内容の変更を行い、サーバに再登録します。

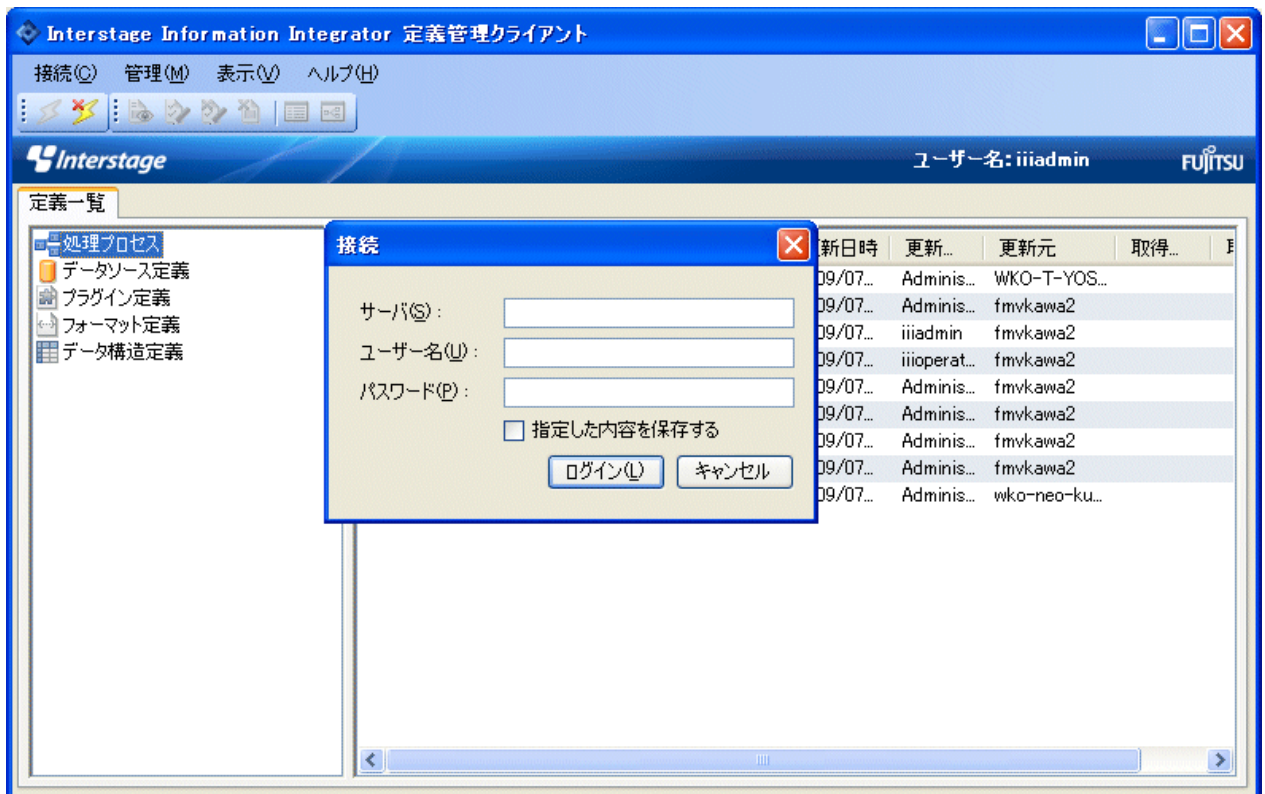
Information Integratorサーバから定義内容を取得する操作手順を以下に説明します。

### デザインシートの取得

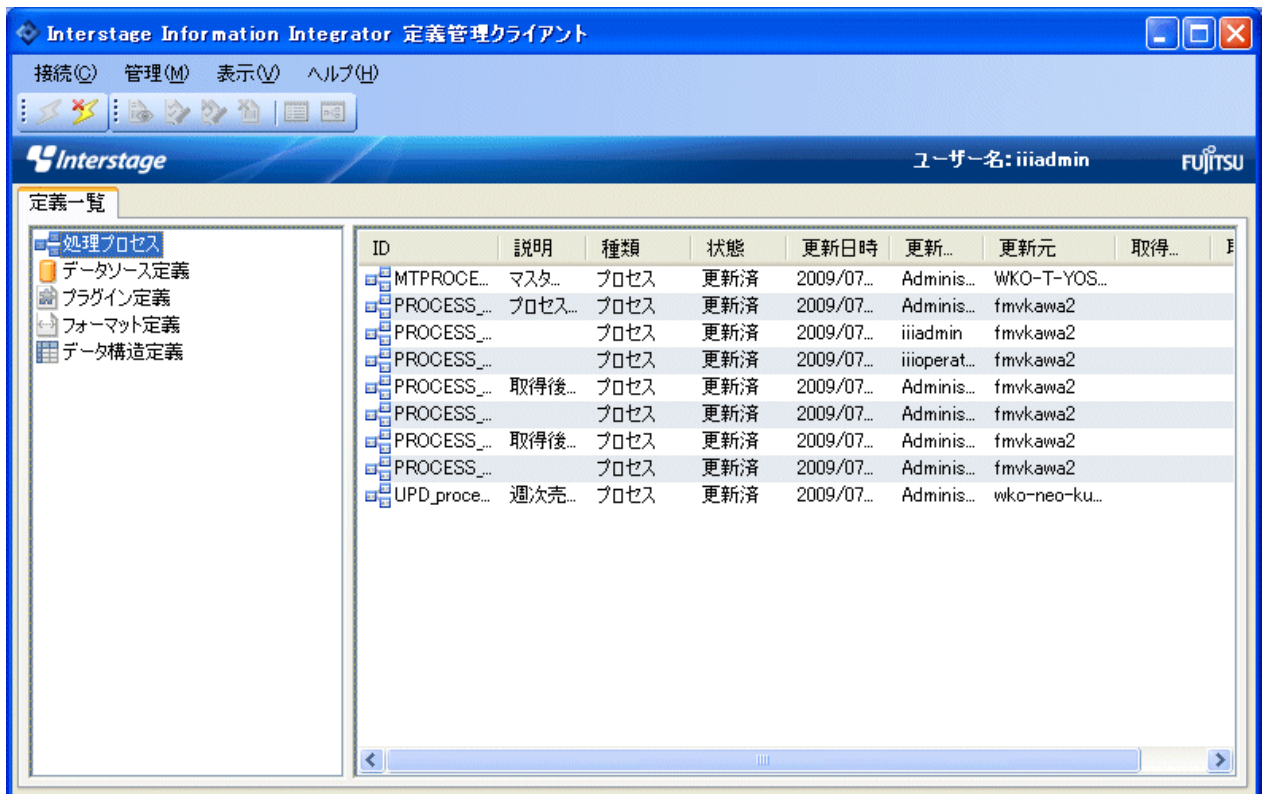
1. 定義管理クライアントを起動します。

[スタート]メニューから[すべてのプログラム]-[Interstage Information Integrator]-[定義管理クライアント]を選択します。

→[接続]画面が表示されます。



2. 接続するユーザー名、パスワードおよびInformation Integratorサーバ名を入力し、[ログイン]ボタンを選択します。  
→Information Integratorサーバに接続されます。



3. デザインシートを取得します。  
右ウィンドウの一覧から、定義内容を変更するデザインシートを選択して、右クリックメニューの[取得]を選択します。  
→取得したデザインシートがExcel上に展開します。

### 注意

サーバに登録した定義内容をデザインシートで取得する際、定義情報だけが取得できます。登録時に作成したデザインシートにExcelの機能で編集したレイアウト情報(列や行の幅、罫線、色設定など)は復元されません。

## 7.1.5 デザインシートの一時保存

作成中のデザインシートは、Excelブックとしてパソコンに一時保存することができます。再度編集する際は、保存したExcelブックをExcelで開くことにより、デザインシートでの設計作業を継続することができます。

### 注意

デザインシートの保存を行う場合、Excelのバージョンに応じて以下の保存形式を選択してください。

Excel 2003の場合:「Microsoft Office Excelブック (\*.xls)」

Excel 2007/2010の場合:「Excelマクロ有効ブック(\*.xlsm)」または「Excel97-2003ブック(\*.xls)」

「\*.xls」形式で保存したデザインシートをExcel 2007/2010で利用する場合、Excelが互換モードで動作します。

## 7.1.6 デザインシートの複写

Excelの機能を利用してデザインシートを複写することで、同じような定義を大量に作成する場合に、設計作業を効率化することができます。

デザインシートは、Excelブック単位での複写と、Excelシート単位での複写が可能です。

複写したデザインシートの定義IDを変更し、定義内容を編集して登録することで、同じ定義の設計を繰り返し行うことなく定義作成が可能になります。

## 7.2 処理プロセスの処理手順の設計

デザインシートを利用した処理プロセスの設計では、プロセス定義シートを利用し、以下を設計します。

- ・ ファンクション処理の処理手順
- ・ 処理プロセスの実行スケジュール

処理プロセスごとにExcelブックを用意してデザインシートを作成します。プロセス定義シートは、1つだけ作成してください。複数のプロセス定義を同じExcelブックに作成することはできません。

複数の処理プロセスを同時に設計する場合には、複数のExcelブックを作成してください。

以下にデザインシート設計の基本操作手順を説明します。

### 1. プロセス定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[処理プロセス]-[プロセス定義]を選択し、プロセス定義シートを作成します。

### 2. ファンクション処理の処理手順

処理プロセスに含まれる各ファンクションを入力し、先行処理の項番を指定することで、ファンクション処理の処理手順を設計します。

### 基本処理

項番	ファンクションID*	処理の種類*	先行処理の項番
1	input_sales_tran	データ収集	
2	input_goods_master	データ収集	
3	bin_goods	データ変換	1,2
4	sales001	データ配付	3
5	sales002	データ配付	3
6	sales003	データ配付	3

### 注意

- プロセス定義には、データ収集およびデータ配付処理を必ず指定してください。
- プロセス定義の先頭処理には、データ収集を指定し、最終処理には、データ配付を指定してください。ただし、処理プラグイン機能を用いた処理については、先頭および最終の処理として指定することができます。
- 処理プラグイン機能を用いた処理を先頭の処理に指定する場合、後続処理に、対応するデータ収集処理を必ず指定してください。また、その場合、後続のデータ収集処理が複数存在する場合は、プラグイン処理を、データ収集処理の数分用意してください。

### 3. プロセスグループの指定

ユーザー設計に基づいて、プロセスグループを指定します。

プロセス定義ID*	PROCESS_B
説明	
プロセスグループID	progrp_a
順序性制御	
実行スケジュール種別	
実行スケジュール(開始)	
リカバリポイント	なし

### 4. 順序性制御の指定とリカバリポイントの指定

同一プロセスについて順序性制御を行うか否かを指定します。また、リカバリポイントを使用するか否かを指定します。

プロセス定義ID*	PROCESS_B
説明	
プロセスグループID	progrp_a
順序性制御	保証する
実行スケジュール種別	
実行スケジュール(開始)	
リカバリポイント	あり

### 5. 処理プロセスの実行スケジュール

処理プロセスの実行スケジュールを設定する場合、実行スケジュールの種別と開始時刻を設定します。

外部アプリケーションからコマンドでプロセスを起動する場合は、設定しません。

プロセス定義ID*	PROCESS_B
説明	
プロセスグループID	progrp_a
順序性制御	保証する
実行スケジュール種別	日次
実行スケジュール(開始)	12:00
リカバリポイント	あり

## ポイント

一定時間間隔繰り返しとイベント監視実行では、スケジュール実行時間帯を事前に設定しておくことができます。例えば、Information Integratorに、60分間隔で繰り返し実行するプロセスや、収集監視実行のプロセスなどがあり、いずれのプロセスも実行開始時間帯を午前9時以降午後5時までと設定しておくことができます。

この設定は、動作環境ファイルに行います。本設定についての詳細は、“セットアップガイド”を参照してください。

## 例

イベント監視を行う場合

設定項目	設定内容
実行スケジュール種別	イベント監視

なお、1つの処理プロセスに、イベント監視対象データと監視対象外データを混在させることはできません。イベント監視を選択した場合は、必ずデータ収集定義の「収集監視」に「あり」と定義してください。

各項目の詳細は、“デザインシートヘルプ”を参照してください。

また、イベント監視を行う場合のデータ収集定義の設定内容については、“7.4.3 収集方法の設計”を参照してください。

## 7.2.1 ファンクション処理間のデータ受け渡しの設計

データ変換処理の入力データ(変換前のデータ)や、データ振分処理の出力結果が複数のデータになる場合に、ファンクション処理の間で受け渡すデータの対応関係を設計する必要があります。

データ変換処理の入力データや、データ振分処理の出力データには、デザインシートでの設計時に自動でデータIDが設定されます。(「変換データID」および「データ振分定義ID」)

ファンクション処理で扱うデータの対応付けは、プロセス定義シートにおいて、「変換データID」および「振分データID」を前後のファンクションに対して指定することにより設計します。

以下に入力例を示します。

### 基本処理

項番	ファンクションID*	処理の種類*	先行処理の項番	変換データID	振分データID
1	INPUT_SALES001	データ収集		indata1	
2	INPUT_SALES002	データ収集		indata2	
3	INPUT_GOODS	データ収集		indata3	
4	JOIN001	データ変換	1,2,3		
5	DIVIDE001	データ振分	4		
6	OUTPUT_SALES_FILEA	データ配付	5		OutData1
7	OUTPUT_SALES_FILEB	データ配付	5		OutData2
8	OUTPUT_SALES_DB	データ配付	5		OutData3

上記は以下の対応付けを行う例です。

- データ収集処理「INPUT\_SALES001」で収集したデータを、データ変換処理「JOIN001」の入力データindata1に対応付け
- データ収集処理「INPUT\_SALES002」で収集したデータを、データ変換処理「JOIN001」の入力データindata2に対応付け
- データ収集処理「INPUT\_GOODS」で収集したデータを、データ変換処理「JOIN001」の入力データindata3に対応付け
- データ振分処理「DIVIDE001」で振り分けたデータOutData1を、データ配付処理「OUTPUT\_SALES\_FILEA」の配付データに対応付け
- データ振分処理「DIVIDE001」で振り分けたデータOutData2を、データ配付処理「OUTPUT\_SALES\_FILEB」の配付データに対応付け
- データ振分処理「DIVIDE001」で振り分けたデータOutData3を、データ配付処理「OUTPUT\_SALES\_DB」の配付データに対応付け

### 注意

- 複数のデータ振分処理を先行処理として選択することはできません。
- データ振分処理の後続処理が異常完了した場合、データ振分処理が再開待ち状態になります。データ振分処理をiffrstrコマンドで再開してください。(プロセスの先頭からの再実行はできません。)コマンドの詳細は、“コマンドリファレンス”を参照してください。

- データ振分処理の後続処理として、Service Integratorのサービスバス連携出力が含まれるプロセス処理においては、データ振分処理を作成しないでください。

## 7.3 処理プロセスのパラメタの設計

処理プロセスのパラメタの設計は、デザインシートで行います。

1つの処理プロセスの中で実行する、収集処理、変換処理、配付処理の各処理の単位を、ファンクション処理と呼びます。これらのファンクション処理の定義を行います。

以下にファンクション処理の種類と、ファンクション処理の処理単位を以下に説明します。

表7.1 ファンクション処理の種類

ファンクション処理の種類	1つのファンクション処理となる処理単位
データ収集処理	同じデータ構造の1つまたは複数データを収集してデータボックスに格納する処理(収集グループ)。
データ変換処理	1つまたは複数のデータを入力データとして変換処理を行い、1つのデータを出力する処理。
データ振分処理	1つのデータを入力データとし、振分条件によってレコードごとに複数データに振り分ける処理。
データ配付処理	データボックスに格納されたデータを、1つ以上の配付先システムに配付する処理(配付グループ)。
プラグイン処理	1つのプラグイン(外部プログラム)を呼び出す処理。

処理プロセスに含める各ファンクション処理を、それぞれ各デザインシートに定義します。

### 7.3.1 デザインシートの種類

処理プロセスのパラメタを設計するためのデザインシートは、以下のとおりです。

表7.2 処理プロセスのパラメタを設計するためのデザインシート

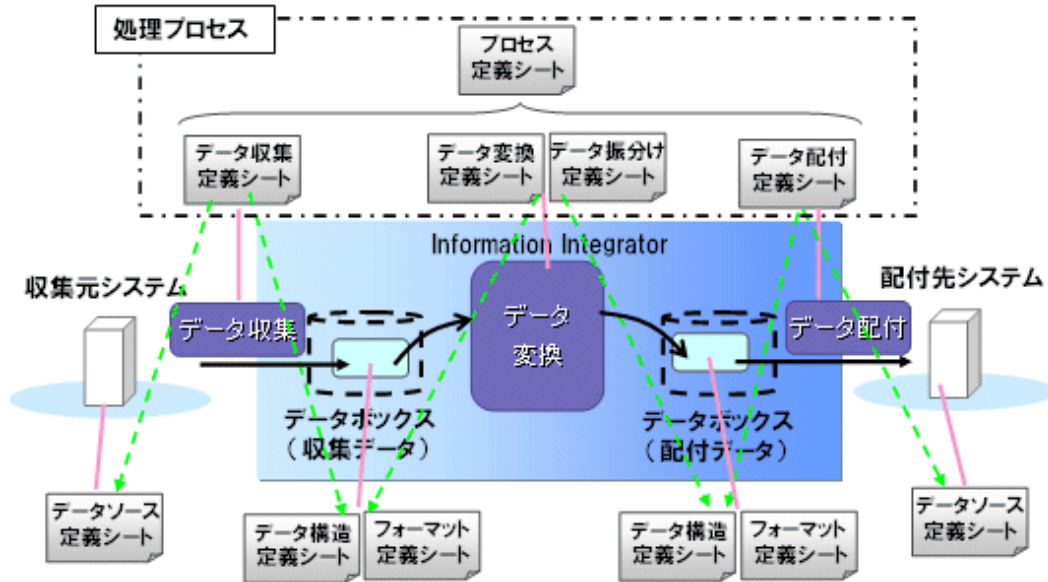
デザインシート名	定義概要
データ収集定義シート	データ収集処理に対して、収集対象とするデータの所在、収集手段を定義します。
データ変換定義シート	データ変換の処理内容(フォーマット変換、項目編集、レコード編集)を定義します。 変換前、変換後のデータに対して、データ構造とフォーマットを定義し、項目の対応付け(マッピング)を定義します。また、演算式を利用した項目ごとの編集、集計やソート、ジョインなどのレコード全体に対する編集を定義します。
データ振分定義シート	データを、振分条件によってレコード毎に複数データに振り分ける処理を定義します。
データ配付定義シート	データ配付処理に対して、配付先の所在、配付手段を定義します。
データソース定義シート	収集先、および配付先のシステム上にあるデータを収集、配付処理を行う上で必要な基本情報(システムの情報、アクセス手段など)を定義します。 データ収集定義シート、データ配付定義シートからIDで参照する情報です。
フォーマット定義シート	データ変換を行う際に、変換前と変換後のデータのフォーマットとして、文字コードやデータ書式などを定義します。
データ構造定義シート	データ変換を行う際の、変換前と変換後のデータの構造として、項目名やデータ属性、桁数などを定義します。
プラグイン定義シート	データ処理の中で呼び出すプラグイン(外部プログラム)の情報を定義します。



処理プロセスに対して、設計する範囲とデザインシートの対応付けを以下の図に示します。(プラグイン定義シートは、図中にありませんが、処理プロセスでプラグイン呼び出しを行う場合に作成します)

点線矢印は、デザインシート間の参照関係を示しています。(例えば、収集元システムに対するデータソース定義シートの情報は、データ収集定義シートから参照します。)

図7.2 設計する範囲とデザインシートの対応付け

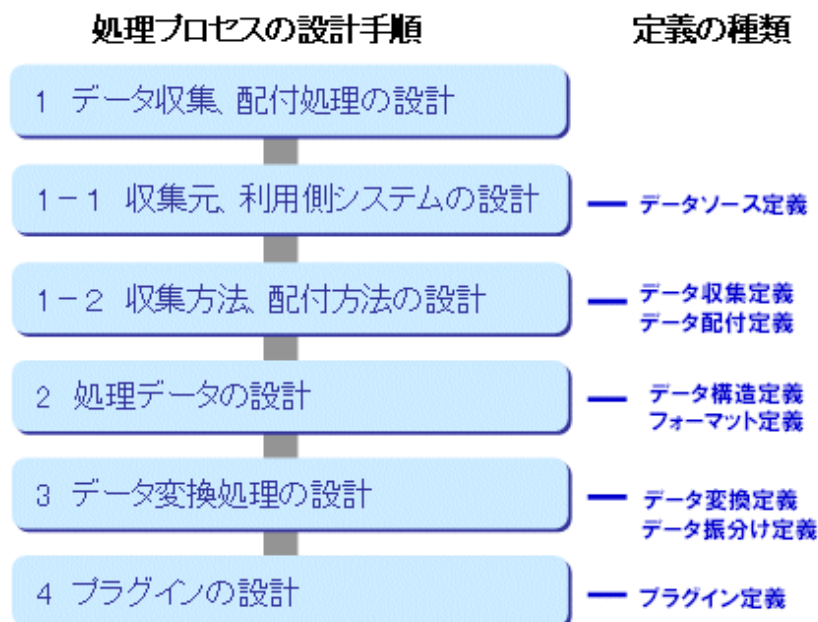


プロセス定義シートは、処理プロセスの処理手順の設計を行うデザインシートです。詳細は、“7.2 処理プロセスの処理手順の設計”を参照してください。

### 7.3.2 処理プロセスのパラメタの設計手順

各デザインシートを作成し、処理プロセスのパラメタの設計を行います。設計の流れと、各処理の設計時に利用するデザインシートを以下に示します。

図7.3 処理プロセスのパラメタの設計手順



各設計手順については、後述を参照してください。

### 7.3.3 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、以下の留意事項があります。

- ・ 見出し行が存在するデータで、データ変換処理またはデータ振分処理を行う場合
  - ー 変換対象のデータを事前に収集するときに(データ収集時)、必ず収集対象のデータが、個別にデータボックスに格納されるようにしてください。  
データ収集定義シートでは、データボックスへの格納方法で「個別型」を指定します。
  - ー 変換対象のデータが一般フォーマットのデータ構造の場合のみ、データ変換処理またはデータ振分処理が可能です。  
データ構造定義シートでは、変換前データの「データ形式」に「一般」を指定します。なお、一般フォーマットについての詳細は、「付録D フォーマット形式」を参照してください。
  - ー 横連結のジョイン(INNER JOIN、OUTER JOIN)は、マスタデータに見出し行が存在するデータは指定できません。  
変換前データの「indata2」以降で設定するフォーマットIDは、フォーマット定義シートの見出し行の有無に「なし」を指定した場合のみ利用可能です。
  - ー 縦連結のユニオン (UNION、UNION ALL)は、できません。
  - ー 変換前データからレコードを抽出する機能(レコード抽出機能)は、できません。

デザインシートについての詳細は、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。

- ・ データ変換処理またはデータ振分処理後に見出し行を付加する場合
  - ー 変換対象のデータが一般フォーマットのデータ構造の場合のみ、データ変換処理またはデータ振分処理が可能です。  
データ構造定義シートでは、変換後データの「データ形式」に「一般」を指定します。なお、一般フォーマットについての詳細は、「付録D フォーマット形式」を参照してください。
  - ー 項目名を元に見出し行を生成します。変換後の文字コードの指定によっては、正しく変換/出力されない場合があります。

## 7.4 データ収集処理の設計

本章では、デザインシートを利用したデータ収集処理の設計手順について説明します。

データ収集処理の設計は以下の手順で行います。

1. 収集元システムの設計
2. 収集データの設計
3. 収集方法の設計

### 7.4.1 収集元システムの設計

データの収集元としてアクセスするシステムの情報(システム名やIPアドレスなど)や、システムへのアクセス手段(データベース情報、ファイル受信のパラメタ、認証情報など)を設計します。

#### データソース定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[データソース定義]を選択し、データソース定義シートを作成します。

- ・ 収集元システムのデータベースからデータを抽出する場合  
「データベース」の行を追加して、収集元システムのデータベースに関するパラメタを入力します。
- ・ 収集元システムのデータをファイル受信で収集する場合  
「ファイル転送」の行を追加して、収集元システムやファイル受信に必要なパラメタを入力します。
- ・ 収集元システムのSalesforceからデータを抽出する場合  
「Salesforce.com」の行を追加して、収集元システムやオブジェクト抽出に必要なパラメタを入力します。



- 収集元システムのInformation Storageからデータを抽出する場合  
「Interstage Information Storage」の行を追加して、収集元システムやIIS抽出に必要なパラメタを入力します。
- 収集元システムのAzure StorageからBLOBデータを収集する場合  
「Azure Storage(BLOB)」の行を追加して、収集元システムやBLOB収集に必要なパラメタを入力します。
- 収集元システムのService Integratorからデータを収集する場合  
「Interstage Service Integrator」の行を追加して、収集元システムやデータ収集に必要なパラメタを入力します。

## ポイント

### Service Integratorからデータを収集する場合

サービスバス連携入力を行う場合は、Service Integrator側の設定が必要です。Service Integratorの設定についての詳細は、“[付録 J サービスバス連携を行う場合のService Integratorの設定](#)”を参照してください。

他の処理プロセスのデータ配付処理や、Information Integratorのデータボックス格納コマンドを利用して、データボックスに格納したデータを利用して処理を行う場合は、データ収集定義シートの「共通」部分にのみパラメタを入力します。

## ポイント

サーバに登録済みのデータソース定義がある場合、データ収集定義シートの「データソースID」に、登録済みのデータソース定義シートの定義IDを指定して、参照利用することが可能です。同じデータソース定義シートを再度作成する必要はありません。

## 7.4.2 収集データの設計

### データボックス名の作成

収集したデータを格納するデータボックス名を決めます。Information Integratorサーバでデータを収集して処理するためには、データボックスが必要です。データボックス名は、データ収集定義シートの「データボックス」に入力します。

データボックス名は、処理プロセス内で一意の値を設定する必要があります。複数プロセスに同じデータボックス名を設定すると、複数プロセス間でデータを受け渡すことができます。

### 収集データのデータ構造、フォーマットの設計

収集してデータボックスに格納するデータのデータ構造、フォーマットを、それぞれデータ構造定義シート、フォーマット定義シートで設計します。

## ポイント

以下の両方を満たす場合、収集データの構造やフォーマットの設計は不要です。

- ファイル形式またはサービスバス連携形式のデータ収集の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、フォーマットの設計は不要です。

- オブジェクト抽出によるデータ収集の場合
- BLOB収集によるデータ収集の場合

## 7.4.3 収集方法の設計

収集方法の設計は、データ収集定義シートを利用して行います。以下に設計手順を説明します。

## データ収集定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[処理プロセス]-[データ収集定義]を選択し、データ収集定義シートを作成します。



例

### イベント監視を行う場合

設定項目	設定内容
運用種別	応答側
収集監視有無	あり
0件データの扱い	正常 イベントを待ち合わせない場合に指定します。
	異常 イベントを待ち合わせる場合に指定します。
収集ファイル名	完全一致、あるいはワイルドカード(*)により指定します。 ディレクトリ監視の場合のみ指定可能です。  ・ワイルドカード指定で、収集するファイル名が「監視対象ディレクトリ名+退避先ディレクトリ名+ファイル名の合計が230バイト」以上の場合 処理ファイル名の指定値に合致していても、常に処理対象外となります。 また、処理プロセスの実行対象外となるファイルの扱いは、動作環境ファイルの「monitorexcept」パラメタの指定に関わらず、常に削除されます。  なお、収集するファイル名が「監視対象ディレクトリ名+退避先ディレクトリ名+ファイル名の合計が230バイト」未満の場合は、ワイルドカード指定の指定値に合致したものを処理対象として動作します。処理プロセスの実行対象外となるファイルの扱いは、動作環境ファイルの「monitorexcept」パラメタの指定に従います。

なお、複数のイベントを監視し、処理プロセスの実行対象となるファイルがすべて到着するまで待ち合わせを行う場合は、イベント監視対象ごとにデータ収集定義を準備してください。1つのデータ収集定義につき、1データソースのみ指定できます。

各項目の詳細は、「デザインシート ヘルプ」を参照してください。

また、イベント監視を行う場合のプロセス定義の設定内容については、「7.2 処理プロセスの処理手順の設計」を参照してください。

#### 「データボックスの格納方法」に「集約型」を指定し、同時に複数のイベントが発生する場合

対応する処理プロセスをイベントの数だけ同時に実行します。タイミングによって初めて実行した処理プロセスでデータを集約して処理し、他の処理プロセスでは0バイトデータとして扱うことになる場合があります。

このような場合は、「0件データの扱い」には「異常」を指定してください。

#### 「0バイトデータの扱い」に「異常」を指定した場合

0バイトデータであるファイルが監視対象ディレクトリに配置され、そのファイルが処理対象ファイルに合致しても対象外ファイルとして処理します。

## データソース定義の参照設定

- ・ 収集元システムのデータベースからデータを抽出する場合

「データベース」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。

- 収集元システムのデータをファイル受信で収集する場合  
「ファイル転送」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 収集元システムのSalesforceオブジェクトを抽出する場合  
「Salesforce.com」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 収集元システムのInformation Storageからデータを抽出する場合  
「Interstage Information Storage」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 収集元システムのAzure Storageからデータを収集する場合  
「Azure Storage(BLOB)」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 収集元システムのService Integratorからデータを収集する場合  
「Interstage Service Integrator」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。

## 収集データの設定

以下の定義IDをそれぞれデータ収集定義シートに記述します。

- 収集データの格納先となる「データボックス」
- 収集データのデータ構造定義の「データ構造定義ID」
- 収集データのフォーマット定義の「フォーマットID」

### ポイント

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- ファイル形式またはサービスバス連携形式のデータ収集の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、「フォーマットID」の指定は不要です。

- オブジェクト抽出によるデータ収集の場合
- BLOB収集によるデータ収集の場合

## 収集処理に必要なパラメタの設計

収集処理に必要なパラメタを、追加した以下の行に設定します。

- 「データベース」
- 「ファイル転送」
- 「Salesforce.com」
- 「Interstage Information Storage」
- 「Azure Storage(BLOB)」
- 「Interstage Service Integrator」

### 注意

Excelファイルを収集する場合

Excelファイルを収集する場合、Excelファイルは、上記の手順でデータボックスへの収集を行います。後続ファンクションでデータ変換やDBなどの連携を行うには、上記の手順での収集処理のあとに、プラグインによるExcelデータからCSVデータへの変換処理が必要です。

Excelファイルで利用するプラグインについての詳細は、「付録K プラグイン機能を利用したExcel連携」を参照してください。

### サービスバス連携入力を行う場合

サービスバス連携入力を行う場合、Service IntegratorのISI運用管理コンソールの表示において、「特定のシーケンス実行に対応する処理プロセスを監視する場合」のInformation Integratorの処理プロセスの見え方がデータ収集定義の「データボックス格納方法」の指定値によって異なることがあります。

異常終了した特定のシーケンス実行に対応する処理プロセスをリカバリする前に、後続の同一シーケンス実行が行われた場合が該当します。

- ・ 「個別型」を指定した場合
  - 特定のシーケンス実行に対応する処理プロセスの履歴には、後続の同一シーケンス実行による処理プロセスの実行結果が表示されます。
  - 後続の同一シーケンス実行に対応する処理プロセスの履歴には、表示すべき実行結果がありません。
- ・ 「集約型」を指定した場合
  - 特定のシーケンス実行に対応する処理プロセスの履歴、および、後続の同一シーケンス実行に対応する処理プロセスの履歴には、後続の同一シーケンス実行による処理プロセスの実行結果が表示されます。

---

## 7.5 データ配付処理の設計

データ配付処理の設計は以下の手順で行います。

1. データの配付先システムの設計
2. 配付データの設計
3. 配付方法の設計

---

### 7.5.1 データ配付先システムの設計

Information Integratorがデータ配付先としてアクセスするシステムの情報(システム名やIPアドレスなど)や、システムへのアクセス手段(データベース情報、ファイル送信のパラメタ、認証情報など)を設計します。

#### データソース定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[データソース定義]を選択し、データソース定義シートを作成します。

- ・ 配付先システムのデータベースを対象にデータを格納する場合  
「データベース」の行を追加して、配付先システムのデータベースに関するパラメタを入力します。
- ・ 配付先システムにデータをファイル送信で配付する場合  
「ファイル転送」の行を追加して、配付先システムの情報やファイル送信に必要なパラメタを入力します。
- ・ 配付先システムのSalesforceへデータを格納する場合  
「Salesforce.com」の行を追加して、収集元システムやオブジェクト格納に必要なパラメタを入力します。
- ・ 配付先システムのInterstage List Creatorで帳票出力を行う場合  
「Interstage List Creator」の行を追加して、配付先システムや帳票出力に必要なパラメタを入力します。
- ・ 配付先システムのInformation Storageへデータを格納する場合  
「Interstage Information Storage」の行を追加して、配付先システムやIIS格納に必要なパラメタを入力します。

- 配付先システムのAzure Storageへデータを格納する場合  
「Azure Storage(BLOB)」の行を追加して、配付先システムやBLOB配付に必要なパラメタを入力します。
- 配付先システムのService Integratorへデータを配付する場合  
配付先システムやService Integrator配付に必要なパラメタを入力します。

## ポイント

### Service Integratorへデータを配付する場合

サービスバス連携出力を行う場合は、Service Integrator側の設定が必要です。Service Integratorの設定についての詳細は、“[付録 J サービスバス連携を行う場合のService Integratorの設定](#)”を参照してください。

前のファンクション処理の出力データに対して、データボックスに格納するだけの処理を行う場合は、データ配付定義シートの「共通」部分にのみパラメタを入力します。

## 注意

サーバに登録済みのデータソース定義がある場合、データ配付定義シートの「データソースID」に、登録済みのデータソース定義の定義IDを指定して参照利用することが可能です。同じデータソース定義を再度作成する必要はありません。

## 7.5.2 配付データの設計

### データボックス名の作成

配付するデータが格納されているデータボックス名を指定します。Information Integratorサーバでデータを配付するためには、データボックスが必要です。データボックス名は、データ配付定義シートの「データボックス」に入力します。

データボックス名は、処理プロセス内で一意の値を設定する必要があります。複数プロセスに同じデータボックス名を設定すると、複数プロセス間でデータを受け渡すことができます。

### 配付データのデータ構造、フォーマットの設計

配付対象のデータに対して、データ構造、フォーマットを設計します。それぞれデータ構造定義シート、フォーマット定義シートで設計します。

## ポイント

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- データベース、オブジェクト以外のデータ配付の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合、「フォーマットID」の指定は不要です。

- オブジェクト格納によるデータ配付の場合
- BLOB配付によるデータ配付の場合

以下の場合、「データ構造ID」の指定は不要です。

- IIS格納によるデータ格納の場合

## 7.5.3 配付方法の設計

配付方法の設計は、データ配付定義シートを利用して行います。以下に設計手順を説明します。

## 配付定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[処理プロセス]-[データ配付定義]を選択し、データ配付定義シートを作成します。

## データソース定義の参照設定

- 配付先システムのデータベースにデータを格納する場合  
「データベース」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付先システムにファイル送信でデータを配付する場合  
「ファイル転送」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付元システムにSalesforceオブジェクトを配付する場合  
「Salesforce.com」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付先システムに帳票データを配付する場合  
「Interstage List Creator」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付元システムにInformation Storage形式のデータを格納する場合  
「Interstage Information Storage」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付先システムにBLOBデータを配付する場合  
「Azure Storage(BLOB)」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。
- 配付先システムにサービスバス連携形式のデータを配付する場合  
「Interstage Service Integrator」の行を追加して、「データソースID」に、利用するデータソース定義の「データソースID」を設定します。

## 配付データの設定

以下の定義IDをそれぞれデータ配付定義シートに記述します。

- 配付データが格納されている「データボックス」
- 配付データのデータ構造定義の「データ構造定義ID」
- 配付データのフォーマット定義の「フォーマットID」

### ポイント

以下の両方を満たす場合、「データ構造定義ID」、「フォーマットID」の指定は不要です。

- データベース、オブジェクト以外のデータ配付の場合
- 収集データに対して変換処理をせずにそのまま配付する処理プロセスの場合

以下の場合は、「フォーマットID」の指定は不要です。

- オブジェクト格納によるデータ配付の場合
- BLOB配付によるデータ配付の場合

以下の場合は、「データ構造ID」の指定は不要です。

- IIS格納によるデータ格納の場合

## 配付処理に必要なパラメタの設計

配付処理に必要なパラメタを、追加した以下の行に設定します。

- 「データベース」

- ・「ファイル転送」
- ・「Salesforce.com」
- ・「Interstage List Creator」
- ・「Interstage Information Storage」
- ・「Azure Storage(BLOB)」
- ・「Interstage Service Integrator」

## 注意

### Excelファイルを配付する場合

Excelファイルを配付する場合、Excelファイルは、上記の手順でデータボックスへの配付を行います。Excelファイルを配付するには、上記の手順での配付処理の前に、プラグインによるCSVデータからExcelデータへの変換処理が必要です。

Excelファイルで利用するプラグインについての詳細は、「付録K プラグイン機能を利用したExcel連携」を参照してください。

## 7.6 処理データの構造・フォーマットの設計

本章では、デザインシートを利用した処理データの構造、フォーマットの設計手順について説明します。

### 7.6.1 処理データの設計概要

処理するデータに対して、データ構造およびフォーマットを設計します。

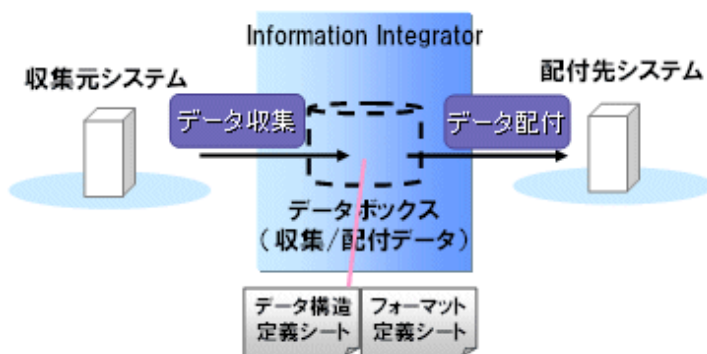
データ構造およびフォーマットの設計が必要となる対象のデータは、処理プロセスの内容によって異なります。処理プロセスのパターンごとに以下に説明します。

#### データ変換処理を含まない処理プロセスの場合

収集データをそのまま配付する場合は、処理データの設計は不要です。

データベースから収集する場合、または収集データに対してデータ構造やフォーマットを変更したデータを配付する場合は、配付するデータの構造、フォーマットを設計します。

図7.4 データ変換処理を含まない場合の設計イメージ



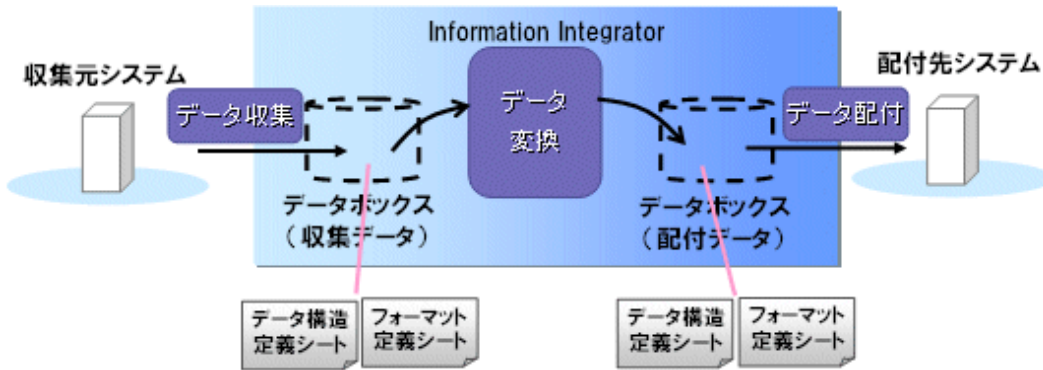
#### 1つだけのデータ変換処理を含む処理プロセスの場合

収集データと配付データのデータ構造およびフォーマットを設計します。

収集データがデータ変換処理の変換前データ、配付データがデータ変換処理の変換後データとなります。



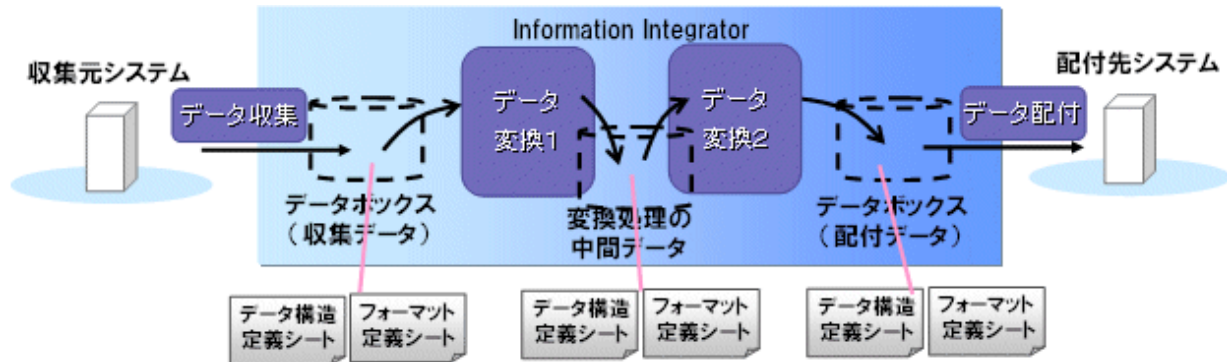
図7.5 1つだけのデータ変換処理を含む場合の設計イメージ



### 複数のデータ変換処理(およびデータ振分処理)を含む処理プロセスの場合

収集データと配信データの他に、データ変換処理の中間結果データに対しても設計を行います。

図7.6 複数のデータ変換処理(およびデータ振分処理)を含む場合の設計イメージ



## 7.6.2 操作手順

以下にデザインシート設計の操作手順を説明します。

### 7.6.3 データ構造の設計

データ構造の設計では、データ構造定義シートを利用し、Information Integratorのデータ変換処理で扱うデータに対して項目の並び順やデータ属性を設計していきます。

#### 1. データ構造定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[データ構造定義]を選択し、データ構造定義シートを作成します。

## 2. データ構造の設計

データの項目名、データ属性、桁数などの情報を入力します。1つのデータ構造定義シートで、1つのデータ構造が設計できます。

	A	B	C	D	E	F	G	H	I
1	データ構造定義								
2									
3	データ構造定義ID *		MO1_GOODS						
4	説明		商品マスタテーブル						
5									
6									
7									
8	データ構造								
9	項番	項目名 *	データ型 *	全体桁 *	小数桁	符号有無	符号位置	符号表	
10	1	品番	SQL_CHAR	15					
11	2	機種名	SQL_CHAR	48					
12	3	商品タイプ	SQL_CHAR	2					
13	4	標準価格	SQL_NUMERIC	15	0				
14	5	製造国コード	SQL_CHAR	10					
15	6	仕入先コード	SQL_CHAR	10					
16	7	仕入先価格	SQL_NUMERIC	32	0				

色のセルを入力できます。  
色のセルは入力できません。  
\* が付いている項目は必須入力です。

メタ収集機能を利用して相手側システムの情報を取り込むときの手順については、後述の“メタ収集機能を利用した設計手順”を参照してください。

### オブジェクト格納時の留意点

オブジェクト格納時の留意点は、それぞれ以下のとおりです。

- データ追加を行うとき

レコード管理IDは、データ追加時にSalesforceで自動採番されます。

そのため、データ構造定義の項目名およびデータに、Salesforceのレコード管理IDを付与する必要はありません。データ構造定義の項目名にレコード管理IDを含めて定義した場合、データ構造定義に定義したレコード管理IDは無効となります。

- データ更新を行うとき

データ構造定義の項目名およびデータに、Salesforceのレコード管理IDを付与してください。

付与したレコード管理IDとSalesforce側のレコード管理IDが一致するレコードのデータが更新されます。レコード管理IDが一致しない場合は、データは更新されません。異常メッセージ出力後、処理は異常終了します。

- データ追加/更新を行うとき

データ構造定義の項目名およびデータに、Salesforceの外部IDを付与してください。

付与した外部IDとSalesforce側の外部IDが一致するレコードのデータが更新されます。外部IDが一致しないレコードは、データが追加されます。



#### レコード管理IDの付与について

データ追加/更新を行うときは、レコード管理IDの定義は不要です。データ構造定義の項目名にレコード管理IDを含めて定義した場合、データ構造定義に定義したレコード管理IDは無効となります。

## 7.6.4 メタ収集機能を利用した設計手順

メタ収集機能を利用して相手側システムの情報を取り込むときの手順について説明します。

収集するメタに応じて、それぞれ説明します。

## CSVファイル

1. [Integratorツール]-[外部メタ読み込み]-[CSVファイル]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(CSV)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)がデータ構造定義シートに表示されます。表示内容を確認してください。

## COPY LIBファイル

COBOLのデータ項目に関する定義(COBOL COPY句)が記述されたファイルのメタ収集定義の手順は、以下のとおりです。

1. [Integratorツール]-[外部メタ読み込み]-[COBOL COPY句ファイル]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(COBOL COPY句)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)がデータ構造定義シートに表示されます。表示内容を確認してください。

## データベース

1. [Integratorツール]-[外部メタ読み込み]-[データベース]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(データベース)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)がデータ構造定義シートに表示されます。  
表示された内容は、Information Integratorサーバで利用可能な属性になっているかなどを確認してください。Information Integratorサーバで利用可能な属性については、“[B.1.4 Information Integratorで収集できるデータベース形式](#)”または“[B.2.4 Information Integratorで配付できるデータベース形式](#)”を参照してください。

## オブジェクト

1. [Integratorツール]-[外部メタ読み込み]-[salesforce.com]を選択します。  
→メタ収集に必要な詳細項目を入力する[外部メタ読み込み(salesforce.com)]画面が表示されます。
2. 選択項目に従って、入力します。各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。  
→収集したメタ情報(項目名や属性情報)がデータ構造定義シートに表示されます。表示内容を確認してください。
3. オブジェクト格納で利用するメタの収集では、オブジェクト格納のアクセス権の有無に関わらずデータ構造定義シートに情報を表示します。アクセス権がない項目は、オブジェクト格納を行うことができないため、アクセス権のない項目をデザインシートから削除してください。デザインシートにアクセス権のない項目が記載された状態でオブジェクト格納を行うとエラーとなります。  
アクセス権についての設定は、“[9.4 オブジェクト抽出・格納利用時のセキュリティ設定情報](#)”を参照してください。  
なお、以下の項目は、オブジェクト格納のアクセス権はありません。必ず削除してください。

- IsDeleted
- CreatedDate
- LastModifiedDate
- SystemModstamp

## 帳票データ

1. [Integratorツール]-[外部メタ読み込み]-[List Creator帳票定義体]を選択します。  
→[ファイルを開く]画面が表示されます。

2. メタの収集元(Interstage List Creatorの帳票定義体)を選択します。

## ポイント

### 帳票定義体について

- 帳票定義体(拡張子.pmd)は、一般的に以下に格納されています。

Interstage List Creator インストール ディレクトリ¥Asset

- 帳票定義体で「集団項目を使用」が指定されている場合、デザインシートの項目には、集団項目を表示します。帳票定義体で「構成項目を使用」が指定されている場合、デザインシートの項目には、構成項目を表示します。

→収集したメタ情報(項目名や属性情報)がデータ構造定義シートに表示されます。表示内容を確認してください。

## ポイント

- 演算結果や定数を変換処理後のデータの項目として出力する場合には、変換後のデータに対してデータ構造を設計する際に、出力項目を含めたデータ構造として設計します。
- 同じようなデータ構造定義を作成する場合には、作成したデータ構造定義シートをExcelの機能で複写し、「データ構造定義ID」を変更して流用することにより、簡単に作成することができます。

## 7.6.5 データフォーマットの設計

データフォーマットの設計では、フォーマット定義シートを利用し、データの形式や文字コードなどの属性を定義します。

### 1. フォーマット定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[フォーマット定義]を選択し、フォーマット定義シートを作成します。

### 2. データフォーマットの設計

1つのデータフォーマット定義シートで複数のフォーマット定義を作成することができます。

## 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、留意事項があります。留意事項については、“[7.3.3 見出し行の存在するデータを扱う場合の留意事項](#)”を参照してください。

## 7.7 データ変換処理の設計

本章では、デザインシートを利用したデータ変換処理の設計手順を説明します。

デザインシートを利用して設計する変換処理には、以下の2種類があります。

### • データ変換処理の設計

1つ、または複数の入力データに対して、項目編集やレコード編集を行い、1つのデータに変換する処理を設計します。データ変換定義シートを利用して設計を行います。

データ変換処理の設計は、以下の手順で行います。

#### — 変換前後のデータ構造およびフォーマットの設計

変換前と変換後のそれぞれのデータ構造やフォーマットを設計することにより変換処理を設計します。

データ構造やフォーマットは、それぞれデータ構造定義シート、フォーマット定義シートで設計を行い、データ変換定義シートで各定義IDを指定することで設計を行います。

変換前後のデータ構造やフォーマットの差異に対して、Information Integratorが必要な変換処理を行います。例えば、変換前のフォーマット定義シートの「文字コード」に「SJMS」、変換後のフォーマット定義シートの「文字コード」に「UTF8」と定義することによって、Shift-JISからUTF8への変換処理を設計できます。

#### 一 項目編集、レコード編集処理の設計

演算式、結合、ソート、または集計などの変換処理を、データ変換定義シートの各フィールドにパラメタ入力することで設計します。1つのデータ変換定義シート上で、複数のデータ変換処理を組み合わせて定義することができます。

#### ・ データ振分処理

1つのデータをレコード条件によって複数のデータに分割する処理、同じデータを複数のシステムに配付する処理を設計します。データ振分定義シートを利用して設計を行います。

## ポイント

### データ変換の処理順序について

ファイル全体のフォーマットに影響するフォーマット変換、文字コード変換は、データ変換の最初に実行します。その次に、項目編集、レコード編集の処理を実行します。集計、ソートはデータ変換の最後に実行します。

一度にすべての変換処理が定義できない場合や、複数の収集データをそれぞれ別の変換処理をした後に、1つのデータに変換する場合などは、データ変換定義シートを複数作成します(複数の変換処理を行う)。

処理プロセスの処理手順設計時に、作成した複数の変換ファンクションの処理順序を決めます。

## ポイント

### 複数のデータ変換定義シートが必要な変換のパターン

以下のような変換処理を設計する場合は、複数のデータ変換定義シートが必要になります。

- ・ ソートや集計の後に項目編集、連結、レコード抽出を実行する場合

## 見出し行の存在するデータを扱う場合の留意事項

データ変換処理またはデータ振分処理で見出し行の存在するデータを扱う場合は、留意事項があります。留意事項については、“7.3.3 見出し行の存在するデータを扱う場合の留意事項”を参照してください。

## 7.7.1 データ変換処理の設計

変換前後のデータに対してデータ構造とフォーマットの設計が完了したら、データ変換定義シートを作成し、データ変換処理の設計を行います。

データ変換処理の設計は、あらかじめ設計したデータ構造定義を読み込んで行うため、データ構造定義シートを利用して変換前と、変換後のデータに対するデータ構造の設計を完了しておく必要があります。

### 7.7.1.1 データ変換定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[処理プロセス]-[データ変換定義]を選択し、データ変換定義シートを作成します。

### 7.7.1.2 データ構造の読み込み

#### データ構造の作成

変換前のデータ(変換処理の入力となるデータ)と変換後のデータ(変換処理の出力となるデータ)に対して、それぞれデータ構造を作成します。変換前のデータが2つ以上の場合は、以下の手順でデータ構造を追加します。



1. データ変換定義シートの任意のセルを選択し、右クリックメニューから[データ構造追加]を選択します。

データ変換定義

データ変換ID*	CONVERT	色のセルを入力できます。
説明		色のセルは入力できません。
エラーデータ		*が付いている項目は必須入力です。
出力ディレクトリ*		

変換動作表示

---

変換前データ

indata1	
データ構造ID*	
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	
レコード	大文字・小文字の取り扱い(半角) 区別する
入力方法	大文字・小文字の取り扱い(全角) 区別する
検索対象外文字	ln
ワード区切り文字	

結合指定行追加  
 結合指定行削除  
 結合キー追加  
 結合キー削除  
**データ構造追加**  
 データ構造削除  
 データ構造読み込み  
 データ構造指定解除  
 演算式追加  
 演算式削除  
 コピー  
 貼り付け

項目	ラベル	項目名	データ型	全体桁	演算式
					演算結果実数

2. 変換前データにデータ構造が1つ追加されます。

データ変換定義

データ変換ID*	CONVERT	色のセルを入力できます。
説明		色のセルは入力できません。
エラーデータ		*が付いている項目は必須入力です。
出力ディレクトリ*		

変換動作表示

---

変換前データ

indata1		結合指定	
データ構造ID*		結合型	
フォーマットID*		キー1	キー2
改行コードの扱い	CRLF		
数値データの属性チェック	しない		
NULL値のチェック	しない		
抽出条件			
レコード	大文字・小文字の取り扱い(半角) 区別する		
入力方法	大文字・小文字の取り扱い(全角) 区別する		
検索対象外文字	ln		
ワード区切り文字			

項目	ラベル	項目名	データ型	全体桁	小数桁	演算式
						演算結果実数

indata2	
データ構造ID*	
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	
レコード	大文字・小文字の取り扱い(半角) 区別する
入力方法	大文字・小文字の取り扱い(全角) 区別する
検索対象外文字	ln
ワード区切り文字	

項目	ラベル	項目名	データ型	全体桁	小数桁	演算式
						演算結果実数

3. 1、2の操作を繰り返し、入力データ数分のデータ構造を追加します。

### データ構造定義の読み込み

作成した入力データに対し、あらかじめ作成したデータ構造定義を、以下の手順で読み込みます。

1. [データ構造ID]に、読み込むデータ構造定義シートの「データ構造ID」を指定します。

2. 入力フィールドを選択し、右クリックメニューの[データ構造読み込み]を選択します。

変換前データ

項目名	値
データ構造ID*	MODEL001
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	
レコード	大文字・小文字の取り扱い(半角) 区別する
入力方法	大文字・小文字の取り扱い(全角) 区別する
検索対象外文字	¥n
ワード区切り文字	

項番	ラベル	項目名	データ型	全体桁	小数桁	演算式	演算結果変数
1-1		機種名	SQL_CHAR	48			
1-2		店舗名	SQL_CHAR	20			
1-3		仕入先名	SQL_CHAR	20			
1-4		配達先	SQL_CHAR	256			
1-5		売単価	SQL_DEC	15	2		
1-6		個数	SQL_DEC	18	0		
1-7		売上高	SQL_DEC	18	2		
1-8		粗利益	SQL_DEC	18	2		

項目名	値
データ構造ID*	MODEL002
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	

右クリックメニュー:

- 結合指定行追加
- 結合指定行削除
- 結合キー追加
- 結合キー削除
- データ構造追加
- データ構造削除
- データ構造読み込み**
- データ構造指定解除
- 演算式追加
- 演算式削除
- コピー
- 貼り付け

ー 指定した[データ構造ID]に該当するデータ構造定義シートが、データ変換定義シートと同じExcelブック上にある場合は、データ構造定義シートの内容から、以下を読み込んで表示します。

- 「項目名」
- 「データ型」
- 「全体桁」
- 「小数桁」

変換前データ

項目名	値
データ構造ID*	MODEL001
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	
レコード	大文字・小文字の取り扱い(半角) 区別する
入力方法	大文字・小文字の取り扱い(全角) 区別する
検索対象外文字	¥n
ワード区切り文字	

項番	ラベル	項目名	データ型	全体桁	小数桁	演算式	演算結果変数
1-1		機種名	SQL_CHAR	48			
1-2		店舗名	SQL_CHAR	20			
1-3		仕入先名	SQL_CHAR	20			
1-4		配達先	SQL_CHAR	256			
1-5		売単価	SQL_DEC	15	2		
1-6		個数	SQL_DEC	18	0		
1-7		売上高	SQL_DEC	18	2		
1-8		粗利益	SQL_DEC	18	2		

項目名	値
データ構造ID*	MODEL002
フォーマットID*	
改行コードの扱い	CRLF
数値データの属性チェック	しない
NULL値のチェック	しない
抽出条件	

ー 指定した「データ構造定義ID」に該当するデータ構造定義シートが、データ変換定義シートと同じExcelブック上にない場合は、以下のメッセージが表示されます。

指定したデータ構造定義がアクティブブック内に存在しません。サーバから取得しますか?

[OK]ボタンをクリックすると、Information Integratorサーバに登録したデータ構造定義を読み込むことができます。



(サーバへ接続していない場合は、[サーバ接続]ダイアログが表示されるので、Information Integratorサーバへ接続してください)

## 注意

- データ構造を読み込んだフィールドに対して編集操作できません。データ構造を変更する場合は、データ構造定義シートを修正し、Information Integratorサーバに登録した後に、再度データ構造を読み込んでください。
- データ構造を再び読み込むと、項目の対応付け(マッピング情報)や演算式情報が削除されるので、再度設定してください。

### 7.7.1.3 データフォーマットの設定

変換前のデータ、変換後のデータそれぞれに対してフォーマットを定義します。「フォーマットID」に、あらかじめ設計したフォーマット定義シートの「フォーマットID」を指定します。

### 7.7.1.4 項目対応付けの設計手順

変換前のデータと変換後のデータに対して、各項目の対応付け(マッピング)を行います。

データ構造を読み込むと、変換前、変換後データの各項目に対して自動で項番が設定されます。

変換後データの設定エリアの「項番(from)」に変換前データの項番を入力し、「項番(from)」と「項番(to)」を並べて設定することにより、変換前後の項目の対応付けを定義します。

以下の対応付けを行う場合の、変換後データの「項番(from)」の入力例を示します。

- 変換前データの1番目から4番目のデータを、同順で変換後データの1番目から4番目に出力
- 変換前データの5番目から8番目のデータを、逆順で変換後データの5番目から8番目に出力

項番(from)	項番(to)	ラベル	項目名	データ型	全長
1-1	1		機種名	SQL CHAR	48
1-2	2		店舗名	SQL CHAR	20
1-3	3		仕入先名	SQL CHAR	20
1-4	4		配送先	SQL CHAR	256
1-8	5		売単価	SQL DECIMAL	15
1-7	6		個数	SQL DECIMAL	18
1-6	7		売上高	SQL DECIMAL	18
1-5	8		粗利益	SQL DECIMAL	18

### 7.7.1.5 項目編集処理の設計手順

各項目に対して演算式を指定します。

編集対象の項目の演算式フィールドに、演算種別、引数を指定します。

## ポイント

データ構造定義のデータ形式が「一般」であり、変換後データの設定で演算式の演算種別に「四則演算」、「文字列の連結」を利用する場合、変換前データの項目を使った演算が可能です。変換前データの項目を使った演算は、ラベル(@n)を利用します。

#### 【設定例】

粗利益 : 小数点以下1桁で切り捨てます。

店舗コード: データの左側の、連続する文字「0」を取り除きます

店舗名 : 半角文字を全角文字に変換します。

[変換前のデータの設定]

項番	ラベル	項目名	データ型	全体桁	小数桁	演算式		
						演算種別	引数	演算結果変数
1-1		粗利益	SQL NUMERIC	10	3	切り捨て	1	
1-2		店舗コード	SQL CHAR	12		トリム	左0	
1-3		店舗名	SQL CHAR	12		全角/半角変換	全角文字	

演算式に利用する項目には、ラベル(@n:nは1から1000までの昇順の数字)を指定し、演算式の引数にラベルを指定します。また、復帰情報番号(¥n:nは1から1000までの昇順の数字)を指定することで、入力データを編集せずに、演算結果を出力データに指定することができます。

#### 【設定例】

売上高と個数のデータに対する四則演算を行います。

売上高のデータに対する編集はせずに、四則演算を行い、結果を出力します。

[変換前のデータの設定]

項番	ラベル	項目名	データ型	全体桁	小数桁	演算式		
						演算種別	引数	演算結果変数
1-1	@1	売上高	SQL DECIMAL	10		四則演算	(@1+100)/@2	¥1
1-2	@2	個数	SQL INTEGER	12				

変換後の出力データの項目の「項目(from)」に演算結果変数「¥1」を指定します。

#### 【設定例】

四則演算の結果を、変換後データの4番目の項目「単価」として出力します。

[変換後データの設定]

項番(from)	項番(to)	ラベル	項目名	データ型	全体桁	小数桁	演算式	
							演算種別	引数
1-1	1		売上高	SQL NUMERIC	10	3		
1-2	2		個数	SQL INTEGER	12			
1-3	3		店舗コード	SQL CHAR	5			
¥1	4		単価	SQL INTEGER				

項目に対して複数の演算を行う場合には、演算式の入力フィールドで、右クリックメニューから[演算式追加]を選択して、演算式の入力フィールドを追加します。

演算結果変数を利用することで、入力データに対して複数の演算結果を出力することが可能です。

#### 【設定例】

仕入先コードの1～3文字と、4～6文字を、それぞれ別項目に出力します。

[変換前データの設定]

項番	ラベル	項目名	データ型	全体桁	小数桁	演算式		演算式			
						演算種別	引数	演算種別	引数		
1-1		売上高	SQL NUMERIC	10	3	切り捨て	1				
1-2		仕入先コード	SQL CHAR	10		文字列の部分抽出	1-3	¥1	文字列の部分抽出	4-6	¥2

[変換後データの設定]

項番(from)	項番(to)	ラベル	項目名	データ型	全体桁	小数桁	演算式		集計	ソート
							演算種別	引数		
1-1	1		売上高	SQL NUMERIC	8	1				
¥1	2		仕入先コード1	SQL CHAR	3					
¥2	3		仕入先コード2	SQL CHAR	3					

#### 【設定例】

名字と名前を連結して、フルネームで出力します。

項番	ラベル	項目名	データ型	全体桁	小数桁
1-1	@1	名字	SQL_CHAR	10	
1-2	@2	名前	SQL_CHAR	10	

項番 (fro)	項番 (to)	ラベル	項目名	データ型	全体桁	小数桁	演算式	
							演算種別	引数
1-1	1		フルネーム	SQL_CHAR	20		文字列の連結	@1+@2

### 【設定例】

販売数と単価から売り上げを計算して出力します。

結合型	INNER JOIN	
キー1	キー2	結合条件
1-1	2-1	完全一致

項番	ラベル	項目名	データ型	全体桁	小数桁
1-1		品名	SQL_CHAR	10	
1-2	@1	販売数	SQL_INTEGER		

項番 (from)	項番	ラベル	項目名	データ型	全体桁	小数桁	演算式	
							演算種別	引数
1-1	1		品名	SQL_CHA	10			
1-2	2		売り上げ	SQL_JNTE			四則演算	@1*@2

項番	ラベル	項目名	データ型	全体桁	小数桁
2-1		品名	SQL_CHAR	10	
2-2	@2	単価	SQL_INTEGER		

### 注意

横連結(INNERJOIN/OUTERJOIN)では、変換前データの各項目を使って演算できます。縦連結(UNION/UNIONALL)では指定できません。

## 7.7.1.6 結合処理の設計手順

以下の手順で、結合指定のフィールドに、結合型、結合キー、結合条件をそれぞれ指定します。

1. 「結合型」に、結合処理の種類を指定します。
2. 結合処理の種類が「UNION ALL」以外の場合は、結合指定行を追加します。  
シート上の任意のセルを選択し、右クリックメニューから[結合指定行追加]を選択します。

**結合指定**

結合型	INNER JOIN	
キー1	キー2	結合条件

**変換後データ**

項番 (from)	項番 (to)	ラベル	
1-1	1		機種名
1-2	2		店舗名
	3		仕入先
	4		配送先
	5		売単価
	6		個数
	7		売上高
	8		粗利益

**outdata1**

データ構造ID *
フォーマットID *
改行コードの扱い
数値データの属性チェック
NULL値のチェック
レコード 統計行
出力 UNIONキー以外

結合指定行追加

- 結合指定行削除
- 結合キー追加
- 結合キー削除

3. 追加された結合指定行に対して、結合キー、結合条件を指定します。

ー INNER JOIN/OUTER JOINの場合

結合する2つの入力データに対して、それぞれ結合キーを指定します。

複数の結合を行う場合は、結合指定行をさらに追加して指定します。

**【設定例】**

以下の結合を行う例を示します。

- 1番目の入力データの2番目の項目と、2番目の入力データの3番目の項目を結合、かつ
- 1番目の入力データの3番目の項目と、3番目の入力データの5番目の項目を結合

**結合指定**

結合型	INNER JOIN	
キー1	キー2	結合条件
1-2	2-3	完全一致
1-3	3-5	完全一致

ー UNIONの場合

重複行を判定する際に対象とする項目を「結合キー」に指定します。

結合キーには、1番目の入力データ(indata1)の項目を指定してください。(例:1-1,1-2)

結合キーは、最大8個まで指定できます。ただし、入力データの項目数が8個以下の場合は、項目数-1個まで指定できます。

ー UNION ALLの場合

結合キーや結合条件の指定は不要のため、結合指定行は指定しません。

### 7.7.1.7 ソート処理の設計手順

変換後のデータとして出力する項目に対して、ソートキーとなる項目を指定します。

ソートの種別(昇順/降順)と、ソートキーの優先順位を組み合わせて設定します。

**【設定例】**

以下の例は、変換後のデータに対して、「1:店舗名を昇順」、「2:機種名を降順」、「3:配送先を昇順」の優先順位で並べ替えて出力する指定です。

**変換後データ**

outdata1		データ構造ID *		フォーマットID *		改行コードの扱い		数値データの属性チェック		NULL値のチェック		レコード		統計行		出力	
		MODEL002				CRLF		しない		しない		なし		統計行		UNIONキー以外の項目	

項番(from)	項番(to)	ラベル	項目名	データ型	全桁桁	小数桁	演算式		集計	ソート
							演算種別	引数		
1-1	1		機種名	SQL CHAR	48					2降順
1-2	2		店舗名	SQL CHAR	20					1昇順
1-3	3		仕入先名	SQL CHAR	20					
1-4	4		配送先	SQL CHAR	256					3昇順
1-5	5		売単価	SQL DECIMAL	15	2				
1-6	6		個数	SQL DECIMAL	18	0				
1-7	7		売上高	SQL DECIMAL	18	2				
1-8	8		粗利益	SQL DECIMAL	18	2				

### 7.7.1.8 集計処理の設計手順

変換後のデータとして出力項目に対して、集計対象とする項目および、集計時にグルーピングする項目を指定します。グルーピングした項目単位で合計を「出力する/しない」を指定できます。

#### 【設定例】

以下の例は、変換後データを、店舗名および機種名でグルーピングを行い、売上高の店舗名ごとの小計、および全体合計を出力する指定です。

変換後データ

項番(from)	項番(to)	ラベル	項目名	データ型	全体桁	小数桁	演算式		集計	ソート
							演算種別	引数		
1-1	1		機種名	SQL CHAR	48				2	
1-2	2		店舗名	SQL CHAR	20				1:小計あり	
1-3	3		仕入先名	SQL CHAR	20					
1-4	4		配達先	SQL CHAR	256					
1-5	5		売単価	SQL DECIMAL	15	2				
1-6	6		個数	SQL DECIMAL	18	0				
1-7	7		売上高	SQL DECIMAL	18	2			グループ計	
1-8	8		粗利益	SQL DECIMAL	18	2				

### 7.7.1.9 変換処理の詳細設定の設計手順

データ変換処理に関する詳細設定について説明します。

データ変換処理実行時の詳細な動作オプションとして、以下のような設定ができます。

#### 【詳細設定の例】

- ・ 変換処理エラー時の動作(処理継続、処理中断)
- ・ 文字コード変換エラー時の代替文字

変換動作定義の設定フィールドにパラメタを入力して設定します。

データ変換定義シートの新規作成時の初期状態では、変換動作定義の設定フィールドは表示されません。

変換動作定義の設定フィールドを表示するには、データ変換定義シートの[変換動作表示]ボタンを選択します。再度非表示にするには、[変換動作表示]ボタンを選択します。

## 7.7.2 データ振分処理の設計

レコード条件で複数のデータに分割するデータ処理は、データ振分処理として設計します。データ振分処理は、データ振分定義シートで設計します。

### 7.7.2.1 データ振分定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[処理プロセス]-[データ振分定義]を選択し、データ振分定義シートを作成します。

### 7.7.2.2 データ振分処理の設計

振分条件が必要な分だけ入力行を追加して、振分条件を指定します。

振分後のデータに対する「振分データID」が自動設定されるので、処理プロセスの処理順序設計時にデータ振分処理の前後のファンクション間でのデータ受渡しの対応関係を設計します。

## ポイント

- 「振分条件」に「other」を指定することができます。「振分条件」に「other」を指定した場合、前述に指定されているすべての振分条件に属さないデータが振分されます。
- 「other」を含めた振分条件の数は、5以内です。

## 注意

- 複数のデータ振分処理を先行処理として選択することはできません。
- 「振分条件」に「other」を指定する場合は、以下を注意してください。
  - 「other」を指定する場合には、1つ以上の振分条件が設定されている必要があります。
  - 「other」は、「振分条件」の最後に指定してください。

## 7.8 プラグイン処理の設計

本章では、デザインシートを利用したプラグイン処理の設計手順について説明します。

### 7.8.1 処理プラグイン機能の設計手順

プラグイン処理(外部プログラムの呼び出し)の設計は、以下の手順で行います。

#### 1. プラグイン定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[プラグイン定義]を選択します。

→プラグイン定義シートが表示されます。「実行形式プラグイン」の行を追加して、プラグインに関する情報を設計します。

各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。

#### 2. プラグイン処理を処理プロセスの手順に追加

処理プロセスの処理順序の設計時に、プラグイン処理を追加します。

基本処理

項番	ファンクションID*	処理の種類*	先行処理の項番
1	input_sales_tran	データ収集	
2	input_goods_master	データ収集	
3	pin_goods	データ変換	1,2
4	plugin001	プラグイン	3
5	sales001	データ配付	4
6	sales002	データ配付	4
7	sales003	データ配付	4

## 注意

プラグイン処理を先頭の処理に指定する場合、後続処理に、対応するデータ収集処理を必ず指定してください。また、その場合、後続のデータ収集処理が複数存在する場合は、プラグイン処理を、データ収集処理の数分用意してください。



### 基本処理

項番	ファンクションID*	処理の種類*	先行処理の項
1	plugin01	プラグイン	
2	plugin02	プラグイン	
3	plugin03	プラグイン	
4	INPUT_SALES001	データ収集	1
5	INPUT_SALES002	データ収集	2
6	INPUT_GOODS	データ収集	3
7	JOIN001	データ変換	4,5,6

## 7.8.2 変換プラグイン機能の設計手順

プラグイン処理(外部プログラムの呼び出し)の設計は、以下の手順で行います。

### 1. プラグイン定義シートの作成

Excelアドインメニュー「Integratorデザインシート」の[定義の新規作成]-[プラグイン定義]を選択します。

→プラグイン定義シートが表示されます。「項目編集プラグイン」の行を追加して、プラグインに関する情報を設計します。

各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。

### 2. プラグイン処理を変換定義シートの処理に追加

変換定義シートの項目別の処理手順に、プラグイン処理を追加します。

#### 変換前データ

indata1			
データ構造ID	MODEL-MTR1_1		
データ形式	一般		
フォーマットID	format_sjis1		
改行コードの扱い	CRLF		
データ チェック	NULLチェック	しない	
	桁溢れチェック	しない	
	桁不足チェック	しない	
	属性チェック	数値データ	しない
		日本語データ	しない
		日付-時間データ	しない
		バイナリデータ	しない
	ビットデータ	しない	
	XMLチェック	しない	
レコード 入力方法	抽出条件		
	大文字・小文字の取り扱い(半角)	区別する	
	大文字・小文字の取り扱い(全角)	区別する	
	検索対象外文字		
	ワード区切り文字		

項番	ラベル	項目名	データ型	全桁桁	小数桁	演算式		
						演算種別	引数	演算結果表数
1-1		コード	SQL_VAR	8				
1-2		商品名	SQL_VAR NCHAR	10		プラグイン	ple01(\$ 商品名" 日本", "20","10" ")	
1-3		価格	SQL_INTE					

各項目の入力内容については、デザインシートの各項目に設定されたコメント、および“デザインシート操作ヘルプ”を参照してください。

## 7.8.3 データを圧縮または解凍する場合

相手側システムから圧縮されたデータを収集したとき、収集したデータを解凍します。また、相手側システムへデータを配付するとき、配付するデータを圧縮します。

データ圧縮およびデータ解凍のプラグインの設計手順は、処理プラグイン機能の設計手順と同じです。詳細は、“7.8.1 処理プラグイン機能の設計手順”を参照してください。



データ圧縮およびデータ解凍を行う場合、プラグイン定義シートの作成時に、データ圧縮コマンドまたはデータ解凍コマンドを指定します。コマンドについては、“[6.10.3.1 ificompcmd\(データ圧縮コマンド\)](#)”および“[6.10.3.2 ifiuncompmd\(データ解凍コマンド\)](#)”を参照してください。

プラグイン定義シートへのコマンドの入力方法は、“[デザインシート操作ヘルプ](#)”を参照してください。

## 7.8.4 ファイル転送エージェントの処理プロセスと連携する場合

---

処理プラグイン機能を利用して、各ファイル転送エージェントの処理プロセスと連携します。処理プラグイン機能の設計手順の詳細は、“[7.8.1 処理プラグイン機能の設計手順](#)”を参照してください。

処理プラグイン機能を利用して、各ファイル転送エージェントの処理プロセスと連携する場合、連携用のコマンドを指定します。コマンドについては、“[6.10.4.1 ificlexerc\(エージェント処理プロセス連携用コマンド\)](#)”を参照してください。

プラグイン定義へのコマンドの入力方法は、“[III Studio ヘルプ](#)”を参照してください。

## 第8章 動作環境の設計

ここでは、Information Integratorの動作環境の構築に関する設計方法について説明します。

### 8.1 多重度の設計について

Information Integratorには、以下の多重度制御機能があります。

- ・ プロセス受付多重度制御
- ・ プロセス実行多重度制御
- ・ データ収集／データ配付／プラグインの各ファンクションの実行多重度制御
- ・ データ変換ファンクションの実行多重度制御
- ・ ファイル送受信実行多重度制御

それぞれの最大多重度は、動作環境ファイルのパラメタでカスタマイズできますが、他の多重度制御のパラメタ値に制約を受けたり、リソース使用量が増加したりする場合があるため、正しく見積りを行った上でカスタマイズする必要があります。

動作環境ファイルについては、“セットアップガイド”を参照してください。

#### 8.1.1 プロセス受付多重度の設計について

処理プロセスのコマンド実行、収集監視による実行、またはスケジュール実行を受付できる最大多重度数を指定します。最大多重度を超える要求は拒否されます。



注意

プロセスの最大実行多重度数より大きな値を指定してください。

#### 8.1.2 プロセス実行多重度の設計について

システムで同時に実行されるプロセスの最大実行多重度数を指定します。最大実行多重度数を超える要求は、「プロセス受付多重度」の範囲内で、多重度数に空きができるまで、Information Integrator内部でキューイングされます。

#### 8.1.3 データ収集／データ配付／プラグインの各ファンクションの実行多重度の設計について

システムで同時に実行されるデータ収集／データ配付／プラグインの各ファンクションの最大実行多重度数を設計します。

利用する機能にあわせて設計します。

- ・ DB格納・DB抽出/HULFTプロトコルによるファイル収集・ファイル配付/オブジェクト抽出/IIS抽出・IIS格納/BLOB収集・BLOB配付/帳票出力/サービスバス連携入力・サービスバス連携出力/処理プラグイン機能を利用する場合

以下のデータソース定義の分類単位で有効になります。ただし、設定値は、すべて共通の値が採用されます。

- － DB連携
- － ファイル転送(HULFTプロトコルのみ)
- － SFDC
- － Interstage Information Storage
- － Interstage Service Integrator
- － Azure Storage
- － 帳票出力
- － プラグイン(実行形式)

例えば、データ収集／データ配付／プラグインの各機能の最大実行多重度を「3」と指定した場合、各機能の処理は、それぞれが最大3多重で実行することができます。

なお、DB連携やファイル転送のように、データ収集とデータ配付を行うような機能の場合は、データ収集とデータ配付の両方の機能での実行数の合計を多重度と考えて、最大実行多重度を設計してください。

例えば、「3」を指定した場合、DB連携では、データ収集・データ配付あわせて最大3多重で実行することができます。

## ポイント

- 最大実行多重度を超える要求は、多重度に空きができるまで、Information Integrator内部でキューイングされます。ただし、データ収集／データ配付機能がHULFTプロトコルによる応答側ファイル転送の場合、最大実行多重度を超える要求を受け付けると要求が拒否されます。
  - 最大実行多重度を1増加させることにより、内部プロセス数は、最大3つ増加します。
- 
- データボックスにあるデータを収集する場合または、データボックスへデータを格納する場合  
最大多重度の設計は不要です。多重度上限はありません。
  - FTP/FTP+/FTPM/HTTP/HTTPS/HICS/SANプロトコルによるファイル収集・ファイル配付  
ファイル収集およびファイル配付を行う場合は、データソース定義の「多重度 - 起動側」「多重度 - 応答側」に指定します。  
詳細は、「[8.1.5 ファイル送受信多重度の設計について](#)」を参照してください。

## 8.1.4 データ変換機能の実行多重度の設計について

システムで同時に実行されるデータ変換およびデータ振分機能の最大実行多重度を指定します。

データ変換機能とデータ振分機能の実行数の和が、最大実行多重度を超える場合は、多重度に空きができるまで、Information Integrator内部でキューイングされます。

ただし、振分条件に「broadcast」を指定したデータ振分機能の場合は、最大実行多重度の指定値とは無関係に動作します。この場合に限り、実行時の上限なく動作します。

## 注意

最大実行多重度を1増加させることにより、内部プロセス数は、最大9つ増加します。

## 8.1.5 ファイル送受信多重度の設計について

### 8.1.5.1 回線の同時使用数の調整

転送処理(ファイル送受信、相手側ジョブ起動)を行うことによるシステムの回線の同時使用数のことを多重度と呼びます。Information Integratorは、この多重度を制限する機能を提供しています。多重度の上限値を設定しない場合、Information Integratorが動作するサーバの性能およびシステム設定により、サーバ自体が不安定になる可能性があります。このため、動作環境ファイルで最大多重度を指定することを推奨します。

動作環境ファイルについては、「[セットアップガイド](#)」を参照してください。

### 多重度のカウントタイミング

多重度は、転送処理のコネクション数でカウントしていません。多重度をカウントアップするタイミングは、転送処理を受け付けた時点です。また、多重度をカウントダウンするタイミングは、転送処理で使用したリソースの開放後です。このため、多重度は、転送処理のリソースを管理する指標となります。

また、転送処理の起動側利用者が転送完了を認識したタイミングでは、応答側は、リソース開放できておらず多重度がカウントダウンできていないことがあります。このタイミングで、起動側から続けて転送要求を発行すると応答側では多重度オーバを検出する可能性があります。このため、多重度の見積り方法としては、起動側:1に対して、応答側:2の採用を推奨します。なお、多重度数が大きい場合は、応答側多重度は、起動側多重度×1.5などの値を採用してください。

## 定義方法

最大多重度の定義は、自システム全体の指定と、相手側システム単位の指定の2段階で行います。相手側システム単位の最大多重度は、起動側多重度と応答側多重度を別に定義します。

自システム全体の最大多重度は、動作環境ファイルの「scale」に、1～256の値の範囲で指定します。相手側システム単位の最大多重度は、データソース定義の「多重度-起動側」、および「多重度-応答側」に、1～256の値の範囲で指定します。

動作環境ファイルについては、“セットアップガイド”を参照してください。

定義の設定項目については、“III Studio ヘルプ”または、“デザインシート操作ヘルプ”を参照してください。

## 多重度の構成

多重度は、以下の2つの項目を足して算出します。

相手側システム数(データソース定義数)やデータ統合処理の内容など、運用方法に合わせて、自システムの多重度を指定します。

- ・ ファイル送受信、相手側ジョブ起動で使用する多重度  
転送処理によって発生する多重度です
- ・ ジョブ結果通知で使用する多重度  
転送処理で起動されたジョブ結果通知を相手側システムに転送するための多重度です

## 多重度の見積り方法

相手側システム(データソース定義)単位で必要となるファイル送受信、相手側ジョブ起動およびジョブ結果通知の最大多重度の見積り方法を以下に示します。

表8.1 多重度の見積り方法

種類	多重度の構成
起動側	(自システムが行う転送処理の多重度)+(相手側システムから受けたジョブ結果通知の多重度)
応答側	(相手側システムから要求される転送処理の多重度)+(自システムが行うジョブ結果通知の多重度)

## 最大多重度を指定した運用の実施

システム全体の多重度は、相手側システム単位の指定した多重度の総和か、またはそれ以下の値を指定することを推奨します。なお、システムおよび相手側システムの多重度が最大多重度を超えると多重度オーバが発生します。この場合は多重度の調整を行ってください。

Information Integratorは、多重度をコネクション単位ではなく、転送時に利用するリソースの獲得開放単位で計算しています。Information Integratorが転送完了通知を行った時点ではコネクションの切断が完了していないため、ファイル送受信で使った多重度は使用中の多重度数に含まれています。なお、多重度が減少するのは、コネクションを切断し、該当するメモリ情報をすべて解放した後になります。

また、最大多重度はCPU、物理メモリなどのサーバ性能や、仮想メモリなどのシステム設定およびInformation Integratorと同時に動作する他のアプリケーションプログラムにより、大きく影響を受けます。Information Integratorおよび運用時同時に動作するアプリケーションプログラムを動作させる運用テストの実施を推奨します。

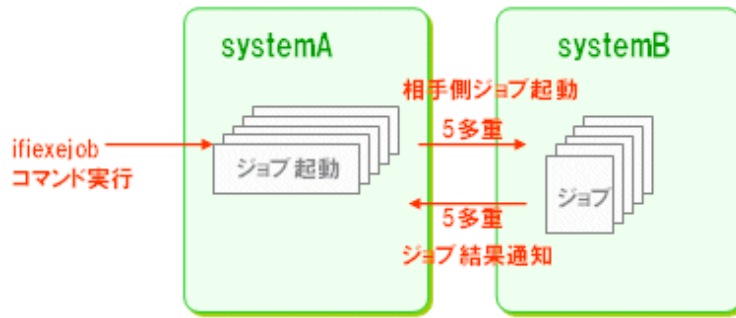
## 多重度オーバ時の動作と多重度の調整方法

ファイル送受信の多重度が最大多重度を超えた場合、多重度オーバにより、ファイル送受信を異常終了します。

### 8.1.5.2 多重度の見積り例(相手側ジョブ起動)

相手側ジョブ起動を行う時に使用する多重度の見積り例は以下のとおりです。以下の例では、systemAからsystemBのジョブを最大5個同時に起動します。また、起動されたジョブはsystemAにジョブ結果通知を行います。

## システムの構成



## 見積り結果

systemA(起動側システム)の場合

多重度-起動側 : 5多重      転送処理(相手側ジョブ起動)

多重度-応答側 : 10多重      制御情報(ジョブ結果通知)

システム全体(scale)の加算値として15多重見積ります。

systemB(応答側システム)の場合

多重度-起動側 : 5多重      制御情報(ジョブ結果通知)

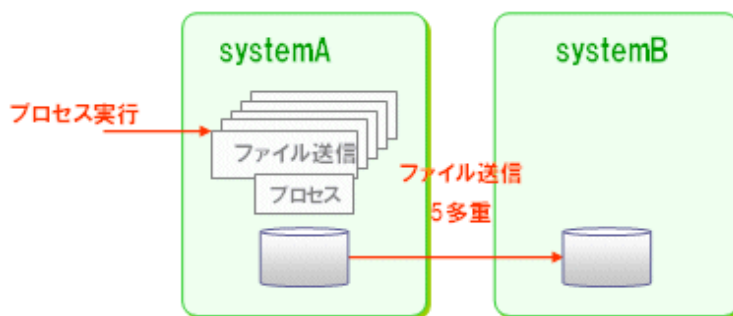
多重度-応答側 : 10多重      転送処理(相手側ジョブ起動)

システム全体(scale)の加算値として15多重見積ります。

### 8.1.5.3 多重度の見積り例(起動側ファイル送信)

起動側システムがファイル送信を行う時に使用する多重度の見積り例は以下のとおりです。以下の例ではsystemAがsystemBに対して最大5多重でファイル送信します。

## システムの構成



## 見積り結果

systemA(起動側システム)の場合

多重度-起動側 : 5多重      転送処理(ファイル送信)

多重度-応答側 : 0多重

システム全体(scale)の加算値として5多重見積ります。

systemB(応答側システム)の場合

多重度-起動側 : 0多重

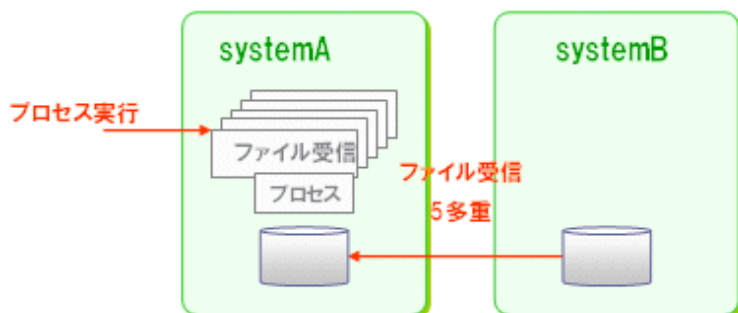
多重度-応答側 : 10多重 転送処理(ファイル送信)

システム全体(scale)の加算値として10多重見積ります。

#### 8.1.5.4 多重度の見積り例(起動側ファイル受信)

起動側システムがファイル受信を行う時に使用する多重度の見積り例は以下のとおりです。以下の例ではsystemAがsystemBから最大5多重でファイル受信します。

##### システムの構成



##### 見積り結果

systemA(起動側システム)の場合

多重度-起動側 : 5多重 転送処理(ファイル受信)

多重度-応答側 : 0多重

システム全体(scale)の加算値として5多重見積ります。

systemB(応答側システム)の場合

多重度-起動側 : 0多重

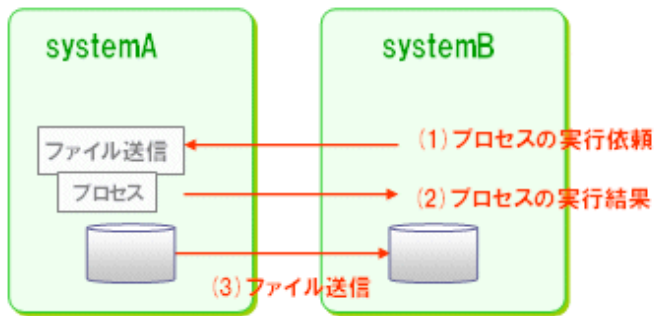
多重度-応答側 : 10多重 転送処理(ファイル受信)

システム全体(scale)の加算値として10多重見積ります。

#### 8.1.5.5 多重度の見積り例(ファイル転送エージェントと連携する場合)

ファイル転送エージェントと連携する場合、内部情報の通信を行うため、ファイル送受信の多重度設計は、内部情報の通信を考慮した設計を行う必要があります。以下に例を示します。

## システムの構成



## 見積り結果

systemA(起動側システム)の場合

多重度-起動側：2多重      転送処理(ファイル送信)、プロセス実行結果通知

多重度-応答側：1多重      プロセス実行要求

システム全体(scale)の加算値として4多重見積ります。

systemB(応答側システム)の場合

多重度-起動側：1多重

多重度-応答側：2多重      転送処理(ファイル受信)、プロセス実行結果通知

システム全体(scale)の加算値として5多重見積ります。

## 8.2 データベースの環境準備

DB抽出およびDB格納を利用する場合に必要なデータベースの環境準備について説明します。

以下の項目について確認してください。各データベースの運用に関する詳細は各データベースのマニュアルを参照してください。

- 各データベースに共通の環境設定
- Symfoware Serverの環境設定
- Oracleの環境設定
- SQL Serverの環境設定
- DB2の環境設定
- PostgreSQLの環境設定
- SQL Azureの環境設定
- その他のデータベース(ODBC接続)の環境設定

また、以下に示す運用に関する留意点も参照してください。

- データベースに共通の留意点
- Symfoware Serverを使用する上での留意点
- Oracleを使用する上での留意点
- SQL Serverを使用する上での留意点
- DB2を使用する上での留意点



- PostgreSQLを使用する上での留意点
- SQL Azureを使用する上での留意点
- その他のデータベース(ODBC接続)を使用する上での留意点

## 8.2.1 各データベースに共通の環境設定

---

DB抽出およびDB格納の機能を利用する場合は、連携データベースの環境設定が必要です。データベースの種別にかかわらず、以下の項目について確認してください。

### データベース

- [データベースの作成]

データベース(Oracleの場合はSID)は、処理を行う前に作成しておく必要があります。すでに作成されているデータベースを利用する場合、あらたにデータベースを作成する必要はありません。

- [データベース名]

データベース名には、各データベースの制限およびDB抽出・格納の制限の両方を満たす名前を使用してください。

### スキーマ

- [スキーマの作成]

#### ー Symfoware Serverに格納する場合

- リモートのSymfoware Serverに格納する場合

スキーマは、処理を行う前に作成しておく必要があります。すでに作成されているスキーマを利用する場合、あらたにスキーマを作成する必要はありません。

- ローカルのSymfoware Serverに格納する場合

ローカルに存在するSymfoware ServerへのDB格納時に限り、スキーマがデータベースに存在しない場合、スキーマを自動的に生成します。スキーマの自動作成機能は、運用テストなどで一時的にスキーマが必要になる場合の利用を推奨します。運用で利用するスキーマについては、DB格納を行う前にデータベースの機能を利用して作成してください。

#### ー Symfoware Server以外に格納する場合

スキーマは、処理を行う前に作成しておく必要があります。すでに作成されているスキーマを利用する場合、あらたにスキーマを作成する必要はありません。

- [スキーマ名]

スキーマ名には、各データベースの制限およびDB抽出・格納の制限の両方を満たす名前を使用してください。

### 表(テーブル)

- [表の作成]

表は処理を行う前に作成しておく必要があります。既に作成されている表に利用する場合、あらたに表を作成する必要はありません。表の定義の方法については、各データベースのマニュアルを参照してください。なお、例外として、ローカルに存在するデータベースへのDB格納時に限り、表がデータベースに存在しない場合、表を自動的に生成します。この場合、利用者はDB格納機能が表を作成するための領域をあらかじめ確保しておく必要があります。表の自動作成機能は、運用テストなどで一時的に表が必要になる場合の利用を推奨します。運用で利用する表については、DB格納を行う前にデータベースの機能を利用して作成してください。

- [表の定義]

データ構造定義の各項目の属性と対応する各列の属性は合致させる必要があります。

- [表名]

表名には、各データベースの制限とDB抽出・格納の制限の両方を満たす名前を使用してください。

- [表のインデックス定義]

DB格納は、格納する表にインデックスを付与しません。利用者がインデックスを事前に作成してください。

## 列(カラム)

- [列名]

列名には、各データベースの制限およびDB抽出・格納の制限の両方を満たす名前を使用してください。

- [列長]

DB抽出で扱えるレコード長の制限はありません。DB格納機能で扱えるレコード長の上限值は32,000バイトです。32,000バイトを超えるレコードは扱えません。レコード長の上限を超えないように設計および運用を行ってください。ただし、31Kバイトを超えるデータのバイナリオブジェクトは長さの総和に加えません。

レコード長は以下の情報から算出可能です。データモデルから算出した以下の値の合計となります。

- 全入力データ項目の属性から算出したデータ長の合計
- NULL値識別用フィールド長の合計

- [列の数および並び]

ローダを利用したDB格納の場合、列の名前と属性を、並び順も含め一致させてください。

その他の場合、データ構造定義の列の数が、表の列数に等しいか、または少ない場合にDB格納またはDB抽出することができません。ただし、データ構造定義の列と同じ名前で、対応する属性をもつ列であることが必要です。なお、データ構造定義の項目の並びと表の列の並びが一致する必要はありません。

## ユーザー

- [ユーザーの作成]

DB抽出・格納で利用するデータベース上のユーザーは、処理を行う前に作成しておく必要があります。すでに作成されているユーザーを利用する場合、新たにユーザーを作成する必要はありません。なお、ユーザーに必要な権限については、“[9.3 DB抽出・DB格納利用時のセキュリティ設定情報](#)”を参照してください。

- [ユーザー名およびパスワード]

ユーザー名およびパスワードには、各データベースの制限およびDB抽出・格納の制限の両方を満たすものを使用してください。

## 文字コード

- DB抽出・格納で、利用可能なデータベースの文字コードについては、“[B.1.4 Information Integrator](#)で収集できるデータベース形式”を参照してください。

## 8.2.2 Symfoware Serverの環境設定

---

DB抽出・格納を利用してSymfoware Serverと連携を行う場合は、以下の項目について確認してください。

### RDBシステム名について

16バイト以内の英数字で指定してください。

### データベース名について

先頭が英字で始まる36文字以内の英数字、または、18文字以内の日本語で指定してください。

### スキーマ名、表名および列名について

スキーマ名、表名および列名の設定は、以下の表を参考にしてください。

表8.2 Symfoware Serverを使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	36文字	36文字	36文字	先頭が英字で始まる英数字です。英字は大文字のみです。
日本語	18文字	18文字	18文字	-----

備考1.英字以外の文字については関知しません。Symfoware Serverの命名規約に従ってください。

備考2.格納時にスキーマおよび表を自動生成する場合、スキーマ名および表名は8文字以内で指定してください。

## ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

なお、SET SYSTEM PARAMETER文でUSER\_CONTROL=YESを指定した環境の場合、以下の注意が必要です。

- Symfoware Serverの認可識別子は、システムに登録している利用者IDと対応させます。
- CREATE USERのWITH句には「OS」を指定してください。

### 【使用例】

```
CREATE USER [システムに登録している利用者ID] WITH OS FOR USER
```

詳細は、Symfoware Serverのマニュアルを参照してください。

なお、CREATE USERのWITH句に「DBMS」を指定して作成した認可識別子でのDB抽出・格納は実行できません。実行した場合は、「DB14031」のメッセージを出力して異常終了します。

Symfoware Serverの認可識別子として登録したIDをデータソース定義のユーザー名に指定してください。

注)USER\_CONTROL=YESを指定すると自動的に、定義済の資源(Symfoware Serverのディクショナリ、データベース、スキーマ、表など)の作成者IDが認可識別子として登録されます。その際、CREATE USER の WITH句には「OS」として登録されます。

## リモート接続について

リモートのSymfoware Serverと連携を行う場合は、Symfoware Serverのクライアント用の動作環境ファイルを作成する必要があります。

クライアント用の動作環境ファイルには、以下の実行パラメタを指定してください。

```
SERVER_SPEC = (通信方法,SQLサーバ名,データ資源名,ホスト名,ポート番号,接続サーバ種別)
```

- 通信方法  
「RDB2\_TCP」を指定してください。
- SQLサーバ名  
任意の識別名を指定してください。SQLサーバ名は、Information Integratorのデータソース定義で必要となります。
- データ資源名  
データベース名を指定してください。
- ホスト名  
ホスト名を指定してください。
- ポート番号  
接続先のSymfoware Serverのポート番号を指定してください。
- 接続サーバ種別  
「OPEN」を指定してください。

クライアント用の動作環境ファイルの設定内容の詳細については、Symfoware Serverのマニュアルを参照してください。

【設定例】

```
SERVER_SPEC = (RDB2_TCP, SQLSV01, DB01, HOST01, 2050, OPEN)
```

### 8.2.3 Oracleの環境設定

DB抽出・格納を利用してOracleと連携を行う場合は、以下の項目について確認してください。

#### SIDについて

先頭が英字で始まる36バイト以内の英数字です。

#### スキーマ名、表名および列名について

スキーマ名、表名および列名について、以下の表に示します。

表8.3 Oracleを使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	30文字	30文字	30文字	先頭が英字で始まる英数字です。英字は大文字のみです。
日本語	15文字	15文字	15文字	-----
英語・日本語混在	15文字	15文字	15文字	-----

備考:英字以外の文字については関知しません。Oracleの命名規約に従ってください。

#### Oracleの予約語の使用について

表名にOracleの予約語を使用しないでください。

#### データ・ディクショナリ・ビューの作成

OracleのUniversal Installerを使わずに初期DBの作成を行っている場合、DB抽出・格納を実行する前に、データ・ディクショナリ・ビューを作成しておく必要があります。データ・ディクショナリ・ビューの作成には、Oracleが提供するCATALOG.SQLスクリプトを実行します。CATALOG.SQLスクリプトについては、Oracleのマニュアルを参照してください。

#### ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

#### ODBCの設定について

Windowsの場合の設定方法については、“8.2.9 ODBCの環境設定について(Windowsの場合)”を参照してください。

UNIXの場合の設定方法については、“8.2.10 ODBCの環境設定について(UNIXの場合)”を参照してください。

#### ALERTファイルの採取

Oracleに対する格納をローダで行う場合、運用中はOracleが出力するALERTファイルを常時採取する様に設定してください。

#### 環境変数「NLS\_LANG」の設定について

Oracleで利用する文字コードによって、Information Integratorサーバの動作するOS上の環境変数「NLS\_LANG」に、Oracleをインストールした時に設定した文字コードを指定してください。

指定できる文字コードは、以下のとおりです。

- Shift-JISの場合
  - JA16SJIS
  - JA16SJISTILDE
  - JA16SJISYEN
- Unicode(UTF8-4)の場合
  - AL32UTF8
- EUC(S90)の場合
  - JA16EUC
  - JA16EUCTILDE
  - JA16EUCYEN

【指定例】文字コード「JA16EUC」を利用する場合

```
NLS_LANG=Japanese_Japan. JA16EUC
```

## 8.2.4 SQL Serverの環境設定

DB抽出・格納を利用してSQL Serverと連携を行う場合は、以下の項目について確認してください。

### インスタンス名について

16バイト以内の英数字または"\_"または"\$"で指定してください。

### データベース名について

36バイト以内の英数字で指定してください。

### スキーマ名、表名および列名について

スキーマ名、表名および列名について、下表に示します。

表8.4 SQL Serverを使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	36文字	36文字	36文字	先頭が英字で始まる英数字です。
日本語	18文字	18文字	18文字	-----
英語・日本語混在	18文字	18文字	18文字	-----

備考:DB抽出・格納は、英字以外の文字については関知しません。SQL Serverの命名規約に従ってください。

### ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

なお、SQL Server認証モード(SQL Server Management Studioのセキュリティ配下のログイン)を使用する場合、使用するログイン名のログインプロパティにおいて、以下の設定が必要です。

- サーバロール
  - sysadminをチェックします。

- ・ 状態  
ログインを[有効]にします。

## ODBCの設定について

SQL Serverと連携を行う場合は、Information Integratorサーバの動作するOS上でODBCデータソースの定義を行い、データベースに接続可能な状態としておく必要があります。

Windowsの場合は、[コントロールパネル]-[管理ツール]-[データソース(ODBC)]を選択して「ODBCデータソースアドミニストレータ」を起動し、システムDSNを作成してください。設定方法の詳細については、ODBCドライバのマニュアルを参照してください。

## 8.2.5 DB2の環境設定

DB抽出・格納を利用してDB2と連携を行う場合は、以下の項目について確認してください。

### インスタンス名について

16バイト以内の英数字または"\_"または"\$"で指定してください。

### データベース別名について

36バイト以内の英数字で指定してください。

### スキーマ名、表名および列名について

スキーマ名、表名および列名について、以下の表に示します。

表8.5 DB2を使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	36文字	36文字	36文字	先頭が英字で始まる英数字です。 Windowsの場合は、英字は大文字のみです。
日本語	18文字	18文字	18文字	-----
英語・日本語混在	18文字	18文字	18文字	-----

備考: 英字以外の文字については関知しません。DB2の命名規約に従ってください。

### ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

## ODBCの設定について

Windowsの場合の設定方法については、“[8.2.9 ODBCの環境設定について\(Windowsの場合\)](#)”を参照してください。

UNIXの場合の設定方法については、“[8.2.10 ODBCの環境設定について\(UNIXの場合\)](#)”を参照してください。

DB抽出・格納を利用してDB2と連携を行う場合は、以下の項目について確認してください。

## 8.2.6 PostgreSQLの環境設定

### データベース名について

36バイト以内の英数字で指定してください。

## スキーマ名、表名および列名について

スキーマ名、表名および列名について、以下の表に示します。

表8.6 PostgreSQLを使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	36文字	36文字	36文字	先頭が英字で始まる英数字です。
日本語	18文字	18文字	18文字	-----

備考: 英字以外の文字については関知しません。PostgreSQLの命名規約に従ってください。

## ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

## ODBCの設定について

設定方法については、“[8.2.10 ODBCの環境設定について\(UNIXの場合\)](#)”を参照してください。

## 8.2.7 SQL Azureの環境設定

---

DB抽出・格納を利用してSQL Azureと連携を行う場合は、以下の項目について確認してください。

### インスタンス名について

16バイト以内の英数字または"\_"または"\$"で指定してください。

### データベース名について

36バイト以内の英数字で指定してください。

### スキーマ名、表名および列名について

スキーマ名、表名および列名について、下表に示します。

表8.7 SQL Azureを使用する場合のスキーマ名、表名および列名

英語・日本語	制限値			備考
	スキーマ名	表名	列名	
英語	36文字	36文字	36文字	先頭が英字で始まる英数字です。
日本語	18文字	18文字	18文字	-----
英語・日本語混在	18文字	18文字	18文字	-----

備考: DB抽出・格納は、英字以外の文字については関知しません。SQL Azureの命名規約に従ってください。

## ユーザー名、パスワードについて

ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

## SQL Azureの認証モードについて

SQL Server認証モードでのみ、Information Integratorから利用可能です。



## ODBCの設定について

SQL Azureと連携を行う場合は、Information Integratorサーバの動作するOS上でODBCデータソースの定義を行い、データベースに接続可能な状態としておく必要があります。

Windowsの場合は、[コントロールパネル]-[管理ツール]-[データソース(ODBC)]を選択して「ODBCデータソースアドミニストレータ」を起動し、システムDSNを作成してください。設定方法の詳細については、ODBCドライバのマニュアルを参照してください。

### ポイント

SQL Azureとの通信は、SSLで行います。通信に必要な以下の設定について確認してください。

- SQL Azureとの接続は、SSL通信で行います。ポート番号「1433」のTCP/IPを使用して接続します。ファイアウォールを透過可能な状態にしてください。
- SQL Azureのファイアウォールの設定では、Information IntegratorサーバのIPアドレスを許可するようにしてください。
- SQL Azureポータルでの暗号化の設定を有効にしてください。

## 8.2.8 その他のデータベース(ODBC接続)の環境設定

DB抽出を利用してその他のデータベース(ODBC接続)と連携を行う場合は、以下の項目について確認してください。

- ODBCの環境設定について  
Windowsの場合の設定方法については、「[8.2.9 ODBCの環境設定について\(Windowsの場合\)](#)」を参照してください。  
UNIXの場合の設定方法については、「[8.2.10 ODBCの環境設定について\(UNIXの場合\)](#)」を参照してください。
- ユーザー名、パスワードについて  
ユーザー名は20文字以内の英数字、パスワードは30文字以内の英数字または記号で指定してください。

## 8.2.9 ODBCの環境設定について(Windowsの場合)

Information Integratorサーバの動作するOS上でODBCデータソースの定義を行い、データベースに接続できる状態にします。

- Windows(64bit)上で32ビット版のInformation Integratorを動作させる場合  
以下の32bit版の「ODBCデータソースアドミニストレータ」を起動し、システムDSNを作成してください。設定方法の詳細については、ODBCドライバのマニュアルを参照してください(32bit版のODBCドライバがインストールされている必要があります)。

```
C:\Windows¥SysWOW64¥odbcad32.exe (WindowsをCドライブにインストールした場合)
```

- 上記以外の場合  
[コントロールパネル]-[管理ツール]-[データソース(ODBC)]を選択して、「ODBCデータソースアドミニストレータ」を起動し、システムDSNを作成してください。設定方法の詳細については、ODBCドライバのマニュアルを参照してください。

### 注意

#### DB2と接続する場合

ODBCデータソース名をデータベース別名と一致させてください。

## 8.2.10 ODBCの環境設定について(UNIXの場合)

Information IntegratorにODBCデータソースの定義を行い、データベースに接続できる状態にします。

### 8.2.10.1 ODBC設定ファイルの編集

以下の手順で、ODBC設定ファイルを編集してください。

1. インストール時に、以下のディレクトリにODBC設定ファイルのサンプルが提供されます。

```
/opt/FJSVifisv/sample/ifiodbcdm.ini
```

2. サンプルを流用して、ODBC設定ファイルを編集します。ODBC設定ファイルのサンプルを任意のフォルダにコピーして、テキストエディタなどを利用して編集してください。
3. 編集が完了したら、ODBC設定ファイルを以下に配置してください。

```
/opt/FJSVifisv/lnk/ETC/ifiodbcdm.ini
```



### 注意

ODBC設定ファイルの配置はInformation Integratorサーバの停止中に実施してください。

## 8.2.10.2 ODBC設定ファイルの記述形式

ODBC設定ファイルに定義するパラメタの記述形式を以下に示します。

表8.8 ODBC設定ファイルの記述形式

記述規約	説明
文字コード	Information Integratorサーバの文字コードにあわせ、「SJIS」、「EUC」、「UTF8」のいずれかで記述してください。
1行の最大文字数	1023バイト
セクション名	行の先頭から[]で囲んで記述します。
キー名	行の先頭から記述します。
キー名ごとのパラメタ	「=」の後ろに記述します。
パラメタの記述	特に断りのない限り、以下の文字を使用してください。 <ul style="list-style-type: none"> <li>・ 英字 : A~Z(大文字)、a~z(小文字)、#、-、@、_(すべて半角文字)</li> <li>・ 数字 : 0~9の半角数字</li> <li>・ 英数字 : 英字および数字</li> <li>・ 日本語 : 全角漢字、全角ひらがな、全角カタカナ、全角英字および全角数字の2バイト文字で構成される文字</li> </ul> ファイル名(パス)には、以下の文字を使用してください。 <ul style="list-style-type: none"> <li>・ 英数字、半角空白、.,, (,)、\、¥、:、/</li> </ul>
スペース/タブ	セクション名、キー名、パラメタ前後の連続する半角空白およびタブ文字は無視されます。
コメント	行頭に「;」または「#」を記述すると、その行はコメントとして扱われます。
改行	1つの定義を複数行にわたって記述することはできません。 なお、改行コードはUNIXでの標準(LFのみ)にしてください。



### 注意

セクション名、キー名、パラメタは、すべて大文字と小文字が区別されます。

## 8.2.10.3 ODBC設定ファイルのセクション名、キー名、パラメタ値

ODBC設定ファイルのセクション名、キー名、パラメタ値について、以下に示します。

表8.9 ODBC設定ファイルのキー名とパラメタ値

セクション名	キー名	説明	指定形式	記述の省略
MAIN	DSN_LIST	ODBCデータソース名(DSN)を指定します。	<p>ODBCデータソース名を指定します(複数ある場合はカンマで区切って列挙します)。</p> <ul style="list-style-type: none"> <li>接続するデータベースがDB2の場合 ODBCデータソース名は、データベース名を指定してください。</li> <li>接続するデータベースがDB2以外の場合 利用可能なODBCデータソース名は、英数字のみです。</li> </ul> <p>「MAIN」はODBCデータソース名として使用できません。</p> <p>ここで指定したODBCデータソース名については、ODBC設定ファイル内にODBCデータソース名のセクションを作成し、必要なキーを設定します。</p>	省略不可
ODBCデータソース名	DESCRIPTI ON	ODBCデータソースの説明を記述します。	任意の説明を記述します。	省略可
	DB	接続先のDB種別を指定します。	<p>以下の形式で指定します。</p> <p>ORACLE: Oracleの場合</p> <p>OTHER: その他のデータベース(ODBC接続)の場合</p>	省略不可
	DRIVER	ODBCドライバの共用オブジェクト(so)ファイルを指定します。	指定するsoファイルについては、各ODBCドライバのマニュアルを参照してください。	省略不可
	TNS	DB種別がOracleの場合のみTNS(Transparent Network Substrate)名を指定します。他のDB種の場合には、指定不要です。	Oracle上で定義したTNS名を指定します。	省略不可
	CON_STR	DB種別がPostgreSQLの場合のみ接続文字列を指定します。他のDB種の場合には、指定不要です。	<p>「パラメタ=パラメタ値;パラメタ=パラメタ値;…」の形式で指定します。以下すべてのパラメタに関する指定をセミコロンで区切って列挙します。</p> <ul style="list-style-type: none"> <li>Server DBMSの動作するサーバ名</li> <li>Port 接続対象となるインスタンスのポート番号</li> <li>Database 利用するデータベース名</li> <li>UID ユーザーID</li> <li>PWD パスワード</li> </ul>	省略不可

## 8.2.10.4 ODBC設定ファイルの記述例

ODBC設定ファイル(32ビット)の記述例を以下に示します。

```
#-----  
# Interstage Information Integrator  
# ODBC initialize file (ifiodbcdm.ini)  
#  
# Copyright FUJITSU LIMITED 2010  
#-----  
[MAIN]  
DSN_LIST=DSN_Oracle11g,DB2DB,DSN_PostgreSQL  
  
[DSN_Oracle11g]  
DESCRIPTION=ODBC Data Source Name for Oracle 11g  
DB=ORACLE  
DRIVER=/opt/oracle/app/product/11.1.0/db_1/lib/libsqora.so.11.1  
TNS=ORCL  
  
[DB2DB]  
DESCRIPTION=ODBC Data Source Name for DB2  
DB=DB2  
DRIVER=/opt/ibm/db2/V9.7/lib32/libdb2.so  
  
[DSN_PostgreSQL]  
DESCRIPTION=ODBC Data Source Name for PostgreSQL  
DB=PostgreSQL  
DRIVER=/usr/lib/psqlodbc.so  
CON_STR=Server=localhost:Database=postgres:UID=postgres:PWD=postgres:Port=5432
```

## 8.2.11 留意事項

### 8.2.11.1 データベースに共通の留意点

DB抽出およびDB格納の機能を利用した運用では、データベースの種別にかかわらず以下の点に留意してください。

#### 入力データの文字コードおよび出力データの文字コードの指定

データボックスの文字コードは、格納先または抽出元のデータベースの文字コードと一致させてください。入力データの文字コード、および出力データの文字コードの指定に誤りがある場合、DB抽出・格納の動作は保証されません。指定できる文字コードについては、「[B.1.4 Information Integratorで収集できるデータベース形式](#)」を参照してください。

なお、指定に誤りがあった場合、以下の現象が発生することがあります。

- DB抽出・格納が出力するメッセージのパラメタが文字化けする
- 抽出または格納されたデータに文字化けがある
- 期待しないスキーマ、表または列に対する処理が行われる

#### 文字型のデータの利用

文字型のデータを利用する場合で、データ内に文字以外の制御コードなどが含まれるときは、フォーマット定義のレコードタイプでバイナリを選択してください。

#### データベースの起動

データベースは処理を行う前に起動しておく必要があります。各データベースで起動方法は異なります。詳細は各データベースのマニュアルを参照してください。

## データベースが作成するログ

データベースが作成するログの領域に関する管理を行いません。データベースのログの領域に関する管理は、利用者が行ってください。ログ機能の詳細は、各データベースのマニュアルを参照してください。

## 同一表への同時格納

同一表に対し、同時に格納処理を行うことはできません。同時に格納処理を行おうとした場合は、後続の要求が異常終了します。

## 格納優先の指定について

以下の条件に該当する場合、DB14101のメッセージを出力してDB格納が異常終了します。

1. 更新格納を実施する場合
  - － 格納先表にプライマリキー、またはユニークキーとなる列が定義されていない場合
  - － 格納先表のプライマリキー、またはユニークキーとなる列に対応する入力データ項目がない場合
  - － 格納先表のプライマリキー、またはユニークキーとなる列に対応する入力データ項目が格納優先として指定されていない場合
2. 格納優先として指定したすべての列がプライマリキー、またはユニークキーの場合

## 抽出方法および格納方法の指定について

抽出方法および格納方法の指定で、アンローダまたはローダを選択した場合、各データベースのアンローダまたはローダの仕様によりDB抽出・格納ができない場合、自動的に動的SQLが選択されます。

## DB抽出のフォーマット定義のレコードタイプについて

DB抽出とデータ変換を1つのプロセス定義で実行する場合、DB抽出のフォーマット定義のレコードタイプには「CSV」を指定することを推奨します。

## DB格納失敗時のロールバックについて

動的SQLによるDB格納の場合、格納時にエラーが発生すると格納開始前の状態にロールバックされます。ローダを選択した場合、ロールバックは保証されません。ロールバックを期待する場合は動的SQLを選択してください。

なお、実行中にキャンセルを行った場合、ロールバックは保証されません。

## シノニムについて

シノニムは利用できません。

## 8.2.11.2 Symfoware Serverを使用する上での留意点

DB抽出・格納を利用してSymfoware Serverと連携を行う場合は、以下の運用方法に関する留意点があります。

### バイナリオブジェクトのDB格納

Symfoware Serverに対してバイナリオブジェクトを格納する場合、以下の点に注意してデータ作成および運用を行ってください。

- ・ 出力先テーブルに31Kバイトを超えるBLOB属性の列を含み、かつ格納方法にローダを指定した場合、データ構造定義と格納先テーブルに定義された列名、列数および列の並びを完全に一致させる必要があります。
- ・ 31Kバイトを超えるSQL\_LONGVARBINARY属性の列を含むテーブルの自動生成は行えません。
- ・ 31Kバイト以上のSQL\_LONGVARBINARY属性の列を含むデータの追加更新、更新はできません。

### 更新でのDB格納を実施する場合

格納対象となるテーブルに最低1個のプライマリキーが必要です。

## DSI分割された表に対するDB格納を行う場合の留意点

DSI範囲外のデータが存在した場合、該当レコードを無視して格納処理を続行します。なお、エラーメッセージは出力されませんので、入力データと格納先表を十分確認の上、実施してください。

## アンローダによるDB抽出について

以下の場合、アンローダによるDB抽出はできません。自動的に動的SQLで抽出を行います。

- ・ 1つ以上の列にNULL制約がある場合、かつ、フォーマット定義の「レコードタイプ」が「バイナリ」の場合
- ・ SQL\_NUMERIC型の列を抽出する場合で、格納先のデータボックスのフォーマット定義が、以下の2つを両方とも満たす指定になっていない場合
  - － 符号位置が先頭(TOP)
  - － 符号表現が文字形式(STR)
- ・ SQL\_DATE型、またはSQL\_TIMESTAMP型の列が含まれる場合

## アンローダ、および、ローダ利用時のデータベースユーザについて

アンローダおよびローダを利用する場合、データベースユーザは、データソース定義で指定したユーザーIDではなく、Information Integratorサーバの特権ユーザーとなります。データベースの権限設定は、Information Integratorサーバの特権ユーザーに対して行ってください。

## 8.2.11.3 Oracleを使用する上での留意点

DB抽出・格納を利用してOracleと連携を行う場合は、以下の運用方法に関する留意点があります。

### DB格納で文字型データにNULL値が含まれる場合

文字型データ中にNULL値が含まれている場合は、フォーマット定義の「レコードタイプ」で「バイナリ」を選択してください。「CSV」を選択した場合、エラーメッセージ「DB14065」が発生します。

### バイナリオブジェクトのDB格納

Oracleに対してバイナリオブジェクトを格納する場合、以下の点に注意してデータ作成および運用を行ってください。

- ・ ローダでの格納はできません。
- ・ 31Kバイトを超えるバイナリオブジェクトを含むデータの追加更新、更新はできません。

### ローダによるDB格納の注意事項

ローダを使用したDB格納を行う場合、以下の点に注意してください。

以下に示す現象を回避するには、格納方法として動的SQLを採用してください。

1. 入力データ項目が整数属性または半語整数属性の場合、かつ入力データに負の最小値(それぞれ-2147483648、-32768)が含まれる場合は、NULL値として格納されます。
2. 入力データ項目が数値属性、かつ入力データに制御文字に該当する値(「0x0A」、「0x0D」など)が含まれる場合、格納処理が異常終了することがあります。
3. UNIXの場合、文字型データ中に制御文字に該当する値「0x0A」が含まれる場合、格納処理が異常終了します。

### Oracleプロセス内スレッドのスケジューリング(Windowsの場合のみ)

WindowsでDB格納を行う場合に、動作が停止し無応答になることがあります。この現象は、Oracleプロセス内のスレッドのWindowsスケジューリング優先順位が低い場合に発生することがあります。上記の現象を回避するには、この優先順位を高くしてください。レジストリの「¥HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Oracle」にある「ORACLE\_PRIORITY」を以下のように設定します。

```
ORACLE_PRIORITY:REG_EXPAND_SZ:CLASS:normal;DEF:highest
```

レジストリの設定については、Windowsのマニュアルを参照してください。

## Oracle 64ビット版を使用する上でのLD\_LIBRARY\_PATHの設定について(Solarisのみ)

32ビットアプリケーションのライブラリを参照してください。

```
LD_LIBRARY_PATH=$ORACLE_HOME/lib32:$LD_LIBRARY_PATH
```

### データベースリンクについて

データベースリンクは利用できません。

### アンローダによるDB抽出について

アンローダによるDB抽出はできません。

## 8.2.11.4 SQL Serverを使用する上での留意点

DB抽出・格納を利用してSQL Serverと連携を行う場合は、以下の運用方法に関する留意点があります。

### バイナリオブジェクトのDB格納

SQL Serverに対してバイナリオブジェクトを格納する場合、以下の点に注意してデータ作成および運用を行ってください。

- ・ 31Kバイトを超えるバイナリオブジェクトを含むデータの追加更新、更新はできません。

### bcpユーティリティを使用する場合のデータベース名、スキーマ名および表名について

SQL Serverの予約語をデータベース名、スキーマ名または表名として使用した場合、bcpユーティリティによるDB抽出およびDB格納はできません。

### bcpユーティリティでのDB格納の動作モードについて

bcpユーティリティは、高速モードまたは通常モードでの格納が可能です。なお、高速モードでの格納では、異常発生時のロールバックは行われません。

以下の条件をすべて満たす場合、高速モードで動作します。

- ・ 格納先表にインデックスが存在しない
- ・ 格納先データベースのselect into/bulkcopyオプションにオンを指定しない

bcpユーティリティの動作モードの切り替えは、DB格納では制御できません。また、DB格納が格納先表を生成する場合、高速モードで動作します。

### DB格納時の文字列属性項目の末尾空白文字列について

文字列属性の入力データ項目の末尾空白文字列の扱いは、SQL Serverの動作に従います。SQL Serverでは、格納先列に対してNULL文字を使用可能としている場合、末尾空白文字列は切り捨てられます。格納先列に対してNULL文字を使用不可能としている場合、末尾空白文字列は格納されます。詳細については、SQL Serverのマニュアルを参照してください。

なお、DB格納が格納先表を生成する場合、格納先列に対するNULL文字の扱いは、格納先データベースの設定に従います。

### DB格納時に表を自動生成する場合のNOT NULL制約について

DB格納が格納先表を生成する場合、格納先列に対するNOT NULL制約の有無は、格納先データベースの設定に依存します。したがって、格納先データベースにNOT NULL制約を付加する設定があった場合、NULL値を含むデータの格納処理は失敗します。詳細については、SQL Serverのマニュアルを参照してください。

### Windows認証が設定されているSQL Serverと連携(格納または抽出)する場合について

SQL Server、およびODBCデータソースのセキュリティ設定で、認証を「Windows認証」に設定してある場合、Information Integratorサーバの動作するOS上に、環境変数「IFIDBSQLSLMODE=Win」を設定してください。





## 注意

この環境変数を設定することで、Information Integratorサーバの特権ユーザーで格納または抽出を実施します。データソース定義で指定したユーザーIDによる格納または抽出は行いません。

### アンローダ(bcpユーティリティ)でのDB抽出について

抽出する列数が多い場合(例:500以上)、DB14204のエラーが発生することがあります。抽出する列数が多い場合は、動的SQLでDB抽出を行ってください。

### アンローダによるDB抽出について

以下の場合、アンローダによるDB抽出はできません。自動的に動的SQLで抽出を行います。

- SQL\_NUMERIC型の列を抽出する場合で、格納先のデータボックスのフォーマット定義に、以下の2つを両方とも満たす指定になっていない場合
  - 符号位置が先頭(TOP)
  - 符号表現が文字形式(STR)
- SQL\_TIMESTAMP型の列が含まれる場合
- フォーマット定義の「レコードタイプ」に「バイナリ形式」が指定されている場合

## 8.2.11.5 DB2を使用する上での留意点

### DB抽出・格納について

DB抽出およびDB格納は、いずれも動的SQLで行います。

## 8.2.11.6 PostgreSQLを使用する上での留意点

### DB抽出・格納について

DB抽出およびDB格納は、いずれも動的SQLで行います。

## 8.2.11.7 その他のデータベース(ODBC接続)を使用する上での留意点

DB抽出を利用してその他のデータベース(ODBC接続)と連携を行う場合は、以下の運用方法に関する留意点があります。

### その他のデータベース(ODBC接続)利用時の動作について

その他のデータベース(ODBC接続)では、指定されたODBCドライバを使ってデータを抽出しますが、期待する動作をするかについては、利用するデータベースとODBCドライバの仕様に依存します。利用する場合には、十分な検証を行った上で、お客様の責任範囲においてご利用ください。

### DB抽出・格納について

DB抽出およびDB格納は、いずれも動的SQLで行います。

### データ格納時のデータの形式について

以下のデータ型のデータを格納するときのデータの形式について説明します。

- SQL\_DATE
- SQL\_TIME
- SQL\_TIMESTAMP
- SQL\_DATE

「YYYY-MM-DD」の形式で格納します。

- SQL\_TIME  
「HH:MM:SS」の形式で格納します。
- SQL\_TIMESTAMP  
「YYYY/MM/DD HH:MM:SS」の形式で格納します。

## 8.3 Interstage List Creatorの環境準備

---

帳票出力を行う場合は、Information Integratorサーバと同一の場所に、Interstage List Creatorのコネクタ機能の設定を行います。

コネクタ機能の設定の詳細および、コネクタ機能以外の必要な環境設定については、Interstage List CreatorのInterstage List Creatorのマニュアルを参照して実施してください。

### ポイント

ローカルのInterstage List Creatorに接続する場合も、Interstage List Creatorと接続するためのコネクタ機能の設定が必要です。

ここでは、コネクタ機能の設定概要について示します。

### Windowsの場合

#### 1. 環境設定バッチの実行

Interstage List Creatorのインストールディレクトリ配下にあるバッチ「setupCT.bat」を実行します。

#### 2. サービス自動起動の設定

Interstage List Creatorを起動したときにコネクタ機能のサービスを自動的に起動するようにする場合は、サービスマネージャから自動起動の設定をお願いします。サービス名は、以下のとおりです。

- ListCREATOR RemotePrint
- ListCREATOR Connector

#### 3. 環境設定

必要に応じて、ポート番号などの環境設定を行います。

## 第9章 セキュリティ設計

Information Integratorのセキュリティについて、以下の項目について説明します。

- ・ 利用者の権限
- ・ ファイル送受信、相手側ジョブ起動利用時のセキュリティ設定情報
- ・ DB抽出・DB格納利用時のセキュリティ設定情報
- ・ Windowsでのセキュリティ注意事項

### 9.1 利用者の権限

Information Integratorの操作に対する利用者の権限について、“[第3章 ユーザー設計](#)”を参照してください

### 9.2 ファイル送受信、相手側ジョブ起動利用時のセキュリティ設定情報

ファイル送受信、および相手側ジョブ起動機能を利用する場合のセキュリティについて以下に説明します。

#### 9.2.1 利用者認証(ログイン検査)

ログイン検査では、以下の契機に通知されるセキュリティパラメタをもとに利用者の認証が行われます。ログイン検査で利用者として認証されなかった場合、ファイル送受信は利用できません。

- ・ 相手側システムからのファイル送信受け
- ・ 相手側システムからのジョブ起動要求受け

#### 9.2.2 相手側ジョブ起動の実行資格

相手側ジョブ起動の実行資格は、ジョブの起動システムに応じて以下の観点で行われます。

- ・ UNIX上の相手側ジョブ起動の場合、ジョブはジョブ起動時に指定したユーザーの権限で起動します。ただし、ユーザーの二次グループは無視されます。
- ・ Windows上の相手側ジョブ起動の場合、Windowsの仕様上、相手側ジョブ起動に必要な利用者の権限は、Administratorsグループに所属している必要があります。Administratorsグループに所属していない場合、メッセージ「TR10325」、「TR20325」、または「TR30325」を出力してジョブの起動に失敗することがあります。また、ドメインを作成している場合、AdministratorsグループをDomain Adminsグループにする必要があります。

なお、ジョブは、相手側ジョブ起動時に指定したユーザーの権限で動作します。ただし、Windows Vista、またはWindows Server 2008上のジョブを起動する場合、起動されたジョブは、システム管理者権限で動作します。

#### 9.2.3 接続システムごとに要求されるセキュリティ情報

Information Integratorの接続する相手側システムの種類によって利用者認証、アクセス権、および起動ジョブの実行資格は、以下のようにチェックされます。相手側システムで、セキュリティ情報を登録しておいてください。

表9.1 接続システムごとのセキュリティ機能

接続する相手側システム	サポートするセキュリティ機能		
	利用者認証(注1)	アクセス権チェック	ジョブの実行資格
グローバルサーバ/ PRIMEFORCE	○	○	△(注2)
UNIX	○	○(注3)	○(注3)
Windows	○	○(注4)	○
その他	相手側システムの機能に依存		

○:サポート △:制限付きサポート

注1)利用者認証は、相手側システムから要求を受けたとき、それ以外の項目は自システムの要求、相手側システムからの要求双方を指します。

注2)JCLの実行資格に依存します。

注3)ユーザーの二次グループに対するサポートは行いません。

注4)ユーザーID権限にローカルログオンの権利を付与する必要があります。

## 9.3 DB抽出・DB格納利用時のセキュリティ設定情報

DB抽出・DB格納利用時のセキュリティについて以下に説明します。

### 実行ユーザーについて

DB抽出・格納では、作業ファイルとデータベースの2種類にアクセスします。これらに対して、それぞれ以下のユーザー権限で動作します。

- ・ 作業ファイル

DB抽出・格納は、OS上、Information Integratorサーバの特権ユーザーの権限で動作します。

- ・ データベース

データベースへのアクセスでは、データソース定義で指定したDBアクセス用セキュリティ情報(ユーザーID、パスワード)を使用します。

ただし、以下の場合、データソース定義で指定したDBアクセス用セキュリティ情報は使用しません。それぞれデータベースへのアクセスは、Information Integratorサーバの特権ユーザーを使用します。

- Symfoware ServerへのDB抽出・格納で、アンローダ、または、ローダを利用する場合
- Windows認証が設定されているSQL ServerへのDB抽出・格納の場合
- PostgreSQLへのDB抽出・格納を行う場合、データベースへのアクセスはODBC設定ファイルの「CON\_STR」で指定されたユーザーID、パスワードを使用します。

### 作業ファイルのアクセス権

作業ファイル作成先のディレクトリに対して書き込み権が必要です。作業ファイルについては、データ収集定義、データ配付定義で指定します。詳細は、“III Studio ヘルプ”または“デザインシート ヘルプ”を参照してください。

### データベースのアクセス権

DB連携機能を使用してデータベースにアクセスするユーザーには、データベースに対するアクセス権として以下の権限を与えてください。

- ・ DB抽出

- 抽出元データベースに対するSELECT ANY DICTIONARY権限(注)
- 抽出元表に対するSELECT権限

- ・ DB格納

- 格納先データベースに対するSELECT ANY DICTIONARY権限(注)
- 格納先データベースに対するCREATE PROCEDURE権限
- 格納先データベースに対するCREATE TABLE権限
- 格納先表に対するSELECT権限
- 格納先表に対するINSERT権限
- 格納先表に対するDELETE権限
- 格納先表に対するUPDATE権限

- 一 格納先表に対するDROP権限

注) 対象データベースがOracleの場合

## 9.4 オブジェクト抽出・格納利用時のセキュリティ設定情報

Salesforceのオブジェクト抽出・格納利用時のセキュリティについて以下に説明します。

### 実行ユーザーについて

Salesforceのオブジェクト抽出・格納では、Salesforceにあるオブジェクトにアクセスします。

Salesforceのオブジェクトへアクセスするユーザー権限は、データソース定義で指定したSalesforceオブジェクトアクセス用セキュリティ情報(ユーザーID、パスワード、セキュリティトークン)を使用してアクセスします。

### オブジェクトのアクセス権

オブジェクトに対するアクセス権として以下の権限を与えてください。

- ・ オブジェクト抽出
  - 一 参照可能(Visible)を指定するなお、「参照のみ(Read-Only)」の指定は任意です。
- ・ オブジェクト格納
  - 一 参照可能(Visible)を指定する
  - 一 参照のみ(Read-Only)は指定しない



参照のみ(Read-Only)を指定すると、アクセスできません。



### 外部IDを利用するときのアクセス権

データ追加/更新を行うときに利用するSalesforceの外部IDは、Salesforce側で外部IDにオブジェクト格納の権限を与えてください。

なお、レコード管理IDは、外部IDとして利用することはできません。レコード管理IDは、オブジェクト格納の権限を与えることができないためです。

## 9.5 帳票出力利用時のセキュリティ設定情報

### 実行ユーザーについて

帳票出力では、作業ファイルおよび帳票出力先にアクセスします。Information Integratorサーバは、作業ファイルおよび帳票出力先へのアクセスは、OS上、Information Integratorサーバの特権ユーザーの権限で動作します。



プリンタへの出力は、Windowsの場合、ListCREATOR RemotePrintのサービスのログオンアカウントの権限で行われます。

プリンタを利用できるユーザーが制限されている場合は、ListCREATOR RemotePrintのサービスのログオンアカウントに設定されているユーザーをプリンタの出力設定に追加(権限付与)してください。

## 作業ファイル出力ディレクトリのアクセス権

作業ファイル作成先のディレクトリに対して書き込み権が必要です。

作業ファイルの出力先の設定については、“セットアップガイド”を参照してください。

## 帳票出力先のアクセス権

帳票をファイルに出力する場合、帳票出力先のディレクトリに対して書き込み権または利用権が必要です。

指定可能な帳票出力先の例は、以下のとおりです。

- ・ 帳票をファイルに出力する場合

帳票出力先のディレクトリに対して書き込み権が必要です。

- ・ 帳票をプリンタに出力する場合

帳票出力先のプリンタを利用する権限が必要です。プリンタに利用者登録を行う必要がある場合などは、実行ユーザーに権限を与える対処が必要です。

指定可能な帳票出力先については、Interstage List Creatorのマニュアルを参照してください。

## 9.6 BLOB収集・配付利用時のセキュリティ設定情報

---

### 実行ユーザーについて

BLOB収集・配付では、Azure StorageにあるBLOBにアクセスします。

BLOBへアクセスするユーザー権限は、データソース定義で指定した「認証キー」を使用してアクセスします。

### BLOBのアクセス権

「private」を指定する場合は、データソース定義で指定した「認証キー」でBLOBが作成されている必要があります。

## 9.7 Windowsでのセキュリティ注意事項

---

以下のWindowsを利用される場合の注意事項を説明します。

- ・ Windows Server 2003
- ・ Windows Server 2008

### ユーザーIDのパスワード設定について

Windowsの設定において、パスワードを設定しないユーザーIDを設定可能です。しかし、本ユーザーIDをInformation Integratorサーバで使用すると、以下のエラーが発生します。Information Integratorサーバを使用する場合は、パスワードを設定したユーザーIDを使用してください。

- ・ III Studio、デザインシート、定義管理クライアントからログオンする場合、指定値の誤りとなります。
- ・ ファイル送受信、または相手側ジョブ起動が認証エラーとなります。

また、Windows Server 2008の場合、ローカルセキュリティポリシーのパスワードポリシーのデフォルトは、「パスワードは、複雑さの要件を満たす必要がある」になっています。この場合、パスワードは英数字と記号の組み合わせが必要となりますが、Information Integratorのパスワードパラメタに指定できる文字に決まりがあるため、注意が必要です。

なお、パスワードパラメタに指定できる文字については、各ヘルプを参照してください。

### Windowsファイアウォールの設定について

以下のサービスパック、OSでは、セキュリティ強化がされています。デフォルトの設定の場合、本製品は影響を受けることがあります。

- ・ Windows Server 2003 Service Pack1以降
- ・ Windows Server 2008

このため、以下の手順でInformation Integratorサーバを「通信を許可するプログラム」としてシステムに登録してください。

1. コントロールパネル内の「Windowsファイアウォール」を実行します。設定パネルが表示されるので[全般]タブの[オン]、または[有効]にチェックが入っていることを確認します。また、[例外を許可しない]、または[すべての着信接続をブロックする]にチェックが入っていないことを確認します。
2. [例外]タブを選択して、[プログラムの追加]ボタンをクリックします。ここで以下のファイルを登録します。
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jndefsv.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jnndisp.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrntdm.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrnspsdm.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrnsphcs.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrncvhs.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrntccntl.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrntftp.exe
  - Information Integratorサーバインストールフォルダ¥lnk¥bin¥f3jntnrnspsftp.exe
3. [例外]タブにて上記ファイル名の左にチェックが入っているのを確認し、設定パネルの[OK]ボタンをクリックします。

## コマンド実行権限について

Windows Server 2008の場合、Information Integratorが提供するすべてのコマンドについて、システム管理者権限に昇格した状態でコマンド投入してください。コマンドプロンプトの起動時に、[管理者として実行]を選択します。「管理者として実行」する方法については、“Windowsヘルプとサポート”を確認してください。

また、システムファイル(hosts、およびservicesファイル)を更新する場合、テキストエディタをシステム管理者権限に昇格する必要があります。



## 第10章 文字コード変換の準備

Information Integratorを使用して異種プラットフォーム間で業務構築を実施する上で必須となる文字コード変換について、設定に関する注意事項を説明します。

### 10.1 文字コード変換機能の環境設定

文字コード変換を行うには、以下の2通りの実行方法があります。

- ・ データ変換定義による設定および処理プロセスの実行
- ・ ユーティリティ(コマンド)の実行

文字コード変換機能における以下の環境設定について説明します。

- ・ エラー発生時の動作(文字コード変換動作環境定義)
- ・ 文字コード変換テーブルの切り替え(ifihschgtblコマンド)
- ・ 文字コード変換テーブルのカスタマイズ(ifihscustコマンド)

#### 10.1.1 エラー発生時の動作

文字コード変換エラーが発生したときの動作を、以下の定義で指定します。

- ・ 文字コード変換動作環境定義(文字コード変換コマンド使用時)
- ・ データ変換定義(プロセス実行時)

主に指定する内容と指定方法について、以下に説明します。

##### 文字コード変換動作環境定義の設定

文字コード変換動作環境定義の設定については、“コマンドリファレンス”を参照してください。

##### データ変換定義の設定

“III Studio ヘルプ”または、“デザインシート操作ヘルプ”を参照してください。

##### 文字コード変換エラー発生時に処理を継続したい場合

Information Integratorサーバでは、文字コード変換できない文字があるとエラーとなり、文字コード変換処理を中断します。しかし、以下の設定を行うことにより、エラーとなった文字を代替文字に変換し、文字コード変換処理を継続させることができます。

- ・ 処理プロセス  
データ変換定義の「文字コード変換エラー発生時の動作」に「継続」を指定します。
- ・ 文字コード変換コマンド使用時  
文字コード変換動作環境定義ファイルに、以下のとおり指定します。

```
CHARACTER_CONVERT_ERROR:CONTINUE
```

##### 文字コード変換エラー発生時の代替文字を変更したい場合

Information Integratorサーバでは、文字コード変換エラー発生時に処理を継続する場合は、エラーとなった文字を代替文字としてアンダースコア(1バイト系文字は「\_」、2バイト系文字は「\_\_」)に変換します。しかし、以下の設定を行うことにより、別の代替文字に変換させることができます。

- ・ 処理プロセス
  - ー 変換先の文字コードがUnicodeでない場合は、データ変換定義の「文字コード変換エラー発生時の代替文字」に1バイト系および2バイト系の代替文字を「XX:YYYY」の形式(16進数表現)で指定します。  
例)1バイト系の代替文字を0x7e、2バイト系の代替文字を0x81a0としたい場合、以下のとおり指定します。

7e:81a0

- 一 変換先の文字コードがUnicodeの場合は、データ変換定義の「Unicode変換エラー発生時の代替文字」に、それぞれ以下のとおり指定します。

- Unicodeの形式がUCS2、またはUTF16の場合

1バイト系および2バイト系の代替文字を以下の形式で指定します

【例】1バイト系の代替文字を0x24、2バイト系の代替文字を0xE080としたい場合、以下のとおり指定します。

```
24:E080
```

XXに指定された24は、U+0024の代替文字になります。

- Unicodeの形式がUTF8の場合

1バイト系および2バイト系の代替文字を以下の形式で指定します

```
XX:TTMMBB(16進数表現)
```

【例】1バイト系の代替文字を0x7f、2バイト系の代替文字を0xC28100としたい場合、以下のとおり指定します。

```
7f:C28100
```

- 文字コード変換コマンド使用時

- 一 変換先の文字コードがUnicodeでない場合は、文字コード変換動作環境定義ファイルに1バイト系および2バイト系の代替文字を以下のとおり指定します。

```
ALTERNATE_CHARACTER1:XX:YYYY(16進数表現)
```

例)1バイト系の代替文字を0x7e、2バイト系の代替文字を0x81a0としたい場合、以下のとおり指定します。

```
ALTERNATE_CHARACTER1:7e:81a0
```

- 一 変換先の文字コードがUnicodeの場合は、文字コード変換動作環境定義ファイルに1バイト系および2バイト系の代替文字を以下のとおり指定します。

- Unicodeの形式がUCS2、またはUTF16の場合

```
UNICODE_ALTERNATE_CHARACTER1:ucs2:XX:YYYY(16進数表現)
```

例)1バイト系の代替文字を0x24、2バイト系の代替文字を0xE080としたい場合、以下のとおり指定します。

```
UNICODE_ALTERNATE_CHARACTER1:ucs2:24:E080
```

- Unicodeの形式がUTF8の場合

```
UNICODE_ALTERNATE_CHARACTER1:utf8:XX:TTMMBB(16進数表現)
```

例)1バイト系の代替文字を0x7f、2バイト系の代替文字を0xC28100としたい場合、以下のとおり指定します。

```
UNICODE_ALTERNATE_CHARACTER1:utf8:7f:C28100
```

## 10.1.2 文字コード変換テーブルの切り替え

JEF-EBCDICの旧字体(‘塚’や‘那’の旧字など)やベンダ固有文字(‘株’や‘㊦’)の文字コード変換でエラーが発生した場合、Information Integratorサーバが使用する文字コード変換テーブルを切り替えることにより、文字コード変換が可能になります。

文字コード変換テーブルの切り替えは、ifihschgtblコマンドで行います。

コマンドの詳細は、“コマンドリファレンス”を参照してください。

ifihschgtblコマンドで切り替えることができる文字コード変換テーブルを以下に示します。

- JEF-EBCDICとShift-JISの変換で使用する変換テーブル

- ・ EUC(U90)とShift-JISの変換で使用する変換テーブル

以下に、文字コード変換テーブル切り替えの例を示します。

### ベンダ固有文字(‘株’や‘㊦’)などを文字コード変換したい場合

ベンダ固有文字を文字コード変換するために、ベンダ固有文字変換登録済テーブルに切り替える必要があります。

1. 以下のコマンドを実行し、現在使用中の文字コード変換テーブルがベンダ固有文字変換未登録テーブルであることを確認します。

```
>ifihshchgtbl -c Jef_Ms -d
hschgtbl:INFO:HS0032: "JEF/EBCDIC" <=> "Sjisms" convert tables are ..
      Table kind [Un-reversible]      Vendor code [undefined]
      ~~~~~
```

Vendor codeは、現在使用中の変換テーブルが以下であることを示します。

Vendor code [undefined]:ベンダ固有文字変換未登録テーブル

Vendor code [definded] :ベンダ固有文字変換登録済テーブル

2. 以下のコマンドを実行し、ベンダ固有文字変換登録済テーブルに切り替えます。

```
>ifihshchgtbl -c Jef_Ms -b
hschgtbl:INFO:HS0033: "JEF/EBCDIC" <=> "Sjisms" convert tables have been changed to ...
      Table kind [Un-reversible]      Vendor code [definded]
      ~~~~~
```

### JEF-EBCDICの旧字体(‘塚’や‘那’の旧字など)を文字コード変換したい場合

1. 以下のコマンドを実行し、現在使用中の文字コード変換テーブルが標準変換テーブル(旧字体を変換しない)であることを確認します。

```
>ifihshchgtbl -c Jef_Ms -d
hschgtbl:INFO:HS0032: "JEF/EBCDIC" <=> "Sjisms" convert tables are ..
      Table kind [Standard]      Vendor code [undefined]
      ~~~~~
```

Table kindは、現在使用中の変換テーブルが以下であることを示します。

Table kind [Standard] :標準変換テーブル(旧字体を変換しない)

Table kind [Un-reversible]:縮退変換テーブル(旧字体を変換する)

2. 以下のコマンドを実行し、縮退変換テーブル(旧字体を変換する)に切り替えます。

```
>ifihshchgtbl -c Jef_Ms -t
hschgtbl:INFO:HS0033: "JEF/EBCDIC" <=> "Sjisms" convert tables have been changed to ...
      Table kind [Un-reversible]      Vendor code [undefined]
      ~~~~~
```



注意

文字コード変換で使用する文字コード変換テーブルは、Information Integratorでいずれか1つになります。

## 10.1.3 文字コード変換テーブルのカスタマイズ

外字の変換規則の変更や、利用者独自の変換を行いたい場合、Information Integratorサーバが使用する文字コード変換テーブルをカスタマイズすることにより、文字コード変換が可能になります。



## 注意

### 文字コード変換テーブルの切り替えとカスタマイズの実施順序

文字コード変換テーブルへのカスタマイズは、文字コード変換テーブルへの切り替えを行った後に対象の文字コード変換テーブルに対して行ってください。

- JefkanaとSjisms間の変換の場合

使用できる文字コード変換テーブルは、以下の4種類です。

- 変換型サポートテーブル・ベンダ固有文字未登録
- 変換型サポートテーブル・ベンダ固有文字登録
- 縮退変換テーブル・ベンダ固有文字未登録
- 縮退変換テーブル・ベンダ固有文字登録

例えば、上記のうち、「変換型サポートテーブル・ベンダ固有文字登録」のカスタマイズを行う場合は、当該テーブルへの切り替えを行った後に行います。

- EUC(U90)系とシフトJIS(MS)系の変換の場合

使用できる文字コード変換テーブルは、以下の2種類です。

- 変換型サポートテーブル
- 縮退変換テーブル

例えば、上記のうち、「変換型サポートテーブル」のカスタマイズを行う場合は、当該テーブルへの切り替えを行った後に行います。

### 10.1.3.1 カスタマイズ方法

文字コード変換テーブルのカスタマイズは、`ifihscust`コマンドで行います。コマンドの詳細は、「コマンドリファレンス」を参照してください。

文字コード変換テーブルをカスタマイズするには、以下の2通りの方法があります。

- 制御ファイルを使用してカスタマイズする方法
- Charset Manager(標準コード変換機能)の文字コード変換表を取り込む方法

#### 制御ファイルを使用してカスタマイズする方法

以下の手順で作業を実施します。

具体例については、「[10.2 制御ファイルを使用したカスタマイズ例](#)」を参照してください。

##### 1. 制御ファイルの作成

制御ファイルを作成してカスタマイズ情報を設定します。制御ファイルの作成方法は、「コマンドリファレンス」を参照してください。

##### 2. カスタマイズ情報の反映

以下のコマンドを実行し、制御ファイルに設定したカスタマイズ情報を、文字コード変換テーブルに反映します。

```
ifihscust -f from_code -t to_code -F filename
```

##### 3. カスタマイズ結果の確認

以下のコマンドを実行し、カスタマイズ結果を確認します。

```
ifihscust -f from_code -t to_code -g start_code
```

#### Charset Managerの文字コード変換表を取り込む方法

以下の手順で作業を実施します。

具体例については、「[10.2 制御ファイルを使用したカスタマイズ例](#)」を参照してください。

#### 1. 標準コード変換機能の文字コード変換表をカスタマイズ

Charset Managerのマニュアルを参照し、標準コード変換機能の文字コード変換表をカスタマイズします。

#### 2. 標準コード変換機能の文字コード変換表の取り込み

以下のコマンドを実行し、標準コード変換機能の文字コード変換表を、文字コード変換テーブルに取り込みます。

```
ifihscust -f from_code -t to_code -d output directory
```

なお、ificmcustコマンドを利用することで、すべての文字コード変換表を1手順で取り込むことができます。コマンドの詳細は、“コマンドリファレンス”を参照してください。

#### 3. カスタマイズ結果の確認

以下のコマンドを実行し、カスタマイズ結果を確認します。

```
ifihscust -f from_code -t to_code -g start_code
```

#### 4. カスタマイズ後に行う定義最適化

(III Studioまたは、デザインシートから実行環境へ定義を登録した後にUnicode系の文字コード変換テーブルのカスタマイズを行った場合のみ)

定義登録後にUnicode系の文字コード変換テーブルのカスタマイズを行った場合は、カスタマイズを行った後に必ず「ifigenoptmz(定義最適化コマンド)」を実行してください。

これは、Information Integrator内部で保持している定義情報(データ構造定義の項目名や抽出条件で指定した文字列など)を正しく評価できなくなり、プロセスの実行に失敗する場合があります。

### 注意

- 標準コード変換機能の文字コード変換表を取り込むことにより、文字コード変換テーブルのカスタマイズ履歴はすべて破棄されます。
- 縮退変換テーブル(旧字体を変換する)を使用している状態で、標準コード変換機能の文字コード変換表を取り込むことはできません。事前に、「ifihschgtbl(文字コード変換テーブル切替え)」コマンドで変換テーブルの使用状態を確認してください。
- カスタマイズは、指定されたコード系に対する、文字コード変換テーブル自身が更新されます。(利用者固有の文字コード対応の定義として別ファイルで、優先的に文字コード変換されるCharset Manager 標準コード変換機能の動作とは異なります。)

## 10.2 制御ファイルを使用したカスタマイズ例

制御ファイルによる文字コード変換テーブルのカスタマイズについて、1バイト系文字の文字コード変換を例にして説明します。

#### 1. 文字コード変換テーブルの確認および変更

ifihschgtblコマンドで文字コード変換テーブルを確認および変更します。

文字コード変換テーブルの種類は以下の4パターンあります。使用する文字コード変換テーブルに合わせて変更してください。

- 縮退変換テーブル(旧字体を変換する)/ベンダ固有文字変換未登録テーブル
- 縮退変換テーブル(旧字体を変換する)/ベンダ固有文字変換登録済テーブル
- 変換型サポートテーブル/ベンダ固有文字変換未登録テーブル
- 変換型サポートテーブル/ベンダ固有文字変換登録済テーブル

コマンドの詳細は、“コマンドリファレンス”の説明を参照してください。

変換型サポートテーブル/ベンダ固有文字変換未登録テーブルを使用の場合の例を以下に示します。

```
#ifihschgtbl -c Jef_Ms -d
ifihschgtbl:INFO:HS0032: "JEF/EBCDIC" <=> "Sjisms" convert tables are ..
Table kind [Un-reversible] Vendor code [undefinded]
```

## 2. 入力文字コードのキーワードと、出力文字コードのキーワードの確認

ifihscustコマンドで入力文字コードのキーワードと、出力文字コードのキーワードを確認します。

「Shift-JIS」から「JEF-EBCDIC(カナ)」字形重視への変換で、入力文字コード「0x21」について確認する例を以下に示します。

```
#ifihscust -f Sjisms -t Jefkana -g 21
#####
# Hscust convert table information
# From Code : Sjisms (0x21 - 0x21)
# To Code : Jefkana
#####

CODE = Sjisms:Jefaugkana

21:4F
```

## 3. 制御ファイルの作成

カスタマイズする情報を設定したファイルを作成します。

ファイル名: cntl\_tbl の例を以下に示します。

```
# cat cntl_tbl
CODE=Sjisms:Jefkana
DUPLICATE=CONVERT
21:5a
5d:6a
7c:4f
```

## 4. 文字コード変換テーブルのカスタマイズ

ifihscustコマンドで、文字コード変換テーブルをカスタマイズします。

「Shift-JIS」から「JEF-EBCDIC(カナ) 字形重視」への文字コード変換テーブルをカスタマイズします。

```
# ifihscust -f Sjisms -t Jefkana -F cntl_tbl
```

## 5. 登録状態を確認

ifihscustコマンドで入力文字コードのキーワードと、出力文字コードのキーワードを確認します。

```
# ifihscust -f Sjisms -t Jefkana -g 21
#####
# Hscust convert table information
# From Code : Sjisms (0x21 - 0x21)
# To Code : Jefkana
#####

CODE = Sjisms:Jefkana

21:5A
```

## 6. 文字コード変換の実行

文字コード変換を実施し、結果を確認します。

## 10.3 Charset Manager文字コード変換表の取り込み例

Charset Manager(標準コード変換機能)の文字コード変換表を取り込むことによる文字コード変換テーブルのカスタマイズについて、ifihscustコマンドを利用した2バイト系文字の文字コード変換を例にして説明します。

「JEF-EBCDIC(カナ) 領域重視」から「Unicode(UTF8)」へ、文字コード変換テーブルを以下のようにカスタマイズする場合の例を以下に示します。

```
JEF 0x67C2 を Unicode(UTF8) 0xEC61
```

### 1. 標準コード変換機能の文字コード変換表のカスタマイズ

対応定義キーワード「JEFKANA UCS2」を指定し、「JEF 0x67C2」を「Unicode 0xEC61」にカスタマイズします。

### 2. 標準コード変換機能の文字コード変換表の取り込み

ifihscustコマンドで、標準コード変換機能の文字コード変換表を、文字コード変換テーブルに取り込みます。

「JEF-EBCDIC(カナ)領域重視」から「Unicode(UTF8)」へ、文字コード変換テーブルをカスタマイズします。

```
# ifihscust -f Jefcorekana -t Unicode -U utf8 -d /temp/output
```

なお、ifihscustコマンドを利用することで、すべての文字コード変換表を1手順で取り込むことができます。コマンドの詳細は、“コマンドリファレンス”を参照してください。

### 3. 登録状態を確認

ifihscustコマンドで入力文字コードのキーワードと、出力文字コードのキーワードを確認します。

```
# ifihscust -f Jefcorekana -t Unicode -U utf8 -g 76C2
#####
# Hscust convert table information
#   From Code : Jefcorekana (0x76C2 - 0x76C2)
#   To   Code : Unicode
#####

CODE = Jefcorekana:Unicode

76C2:EC61
```

### 4. 文字コード変換の実行

文字コード変換を実施し、結果を確認します。

# 付録A 各種の利用可能機能一覧

## A.1 通信プロトコルごとの利用可能機能一覧

ここでは、通信プロトコルごと、抽出・格納対象のデータベースごとの利用可能機能について一覧で表しています。

ファイル送受信、およびジョブ起動で利用する通信プロトコルごとの利用可能機能を以下の表に示します。

表A.1 通信プロトコルごとの利用可能機能一覧

機能大分類	機能小分類	通信プロトコル							
		FTP+	FTP	FTPM	HTTP	HTTPS	HICS (FNA)	SAN	HULFT
ファイル受信	ファイル受信(自局主導)	○	○	○	○	○	○	○	○
	ファイル受信(他局主導)	○	○	○	○	○	○	○	○
ファイル送信	ファイル送信(自局主導)-同報配信	○	○	○	○	○	○	○	○
	ファイル送信(自局主導)-一括配付	○	○	○	○	○	○	○	○
	ファイル送信(自局主導)-個別配付	○	○	○	○	○	○	○	○
	ファイル送信(他局主導)	○	○	○	○	○	○	○	○
ジョブ起動	相手側ジョブ起動	○	×	○	○	○	○	×	×
	ジョブ結果通知	○	×	×	○	○	×	×	×
オプション機能	自動圧縮・自動解凍	○	×	×	×	×	○	×	×
	 <b>ポイント</b> ..... 通信プロトコルがFTP+以外の場合かつ、データ形式がzip形式の場合、Information Integratorが提供するプラグインを利用して、圧縮・解凍を利用することができます。 プラグインの詳細は、“6.10.3 データを圧縮または解凍する場合”を参照してください。 .....								
	転送速度制御	○	○	○	○	○	○	×	×
	ファイル置き換え	○	○	○	○	○	○	○	○(注)
	ファイル追加書き	○	○	○	×	×	○	○	○(注)
	バイナリ転送モード	○	○	○	○	○	○	○	○(注)
	テキスト転送モード	○	○	○	×	×	○	○	○(注)
回線制御	多重度制御	○	○	○	○	○	○	○	○(注)
	コネクション継続	○	×	×	×	×	○	×	×
高信頼性	送達確認	○	×	×	×	×	×	○	×
環境	IPv6アドレス	○	○	○	○	○	—	—	○(注)
	IPv4アドレス	○	○	○	○	○	—	—	○

○:利用可

×:利用不可

—:対象外



注:HULFT側の設定で利用可能



- Linux(32ビット動作環境)では、HICS(FNA)、SANは、利用できません。
- Linux(64ビット動作環境)では、HICS(FNA)は、利用できません。

## A.2 データベースごとの利用可能機能一覧

データベースごとのDB抽出・格納で利用できる機能を、以下の表に示します。

### DB抽出

表A.2 データベースごとの利用可能機能一覧(DB抽出の場合)

	Information Integratorサーバのプラットフォーム	データベース種別	機能	
			全件抽出	条件抽出
自システムのデータベースから抽出	Windows	Symfoware Server	◎	○
		Oracle	○	○
		SQL Server	◎	○
		DB2	○	○
		その他のデータベース(ODBC接続)	○	○
	Solaris	Symfoware Server	◎	○
		Oracle	○	○
		DB2	○	○
		その他のデータベース(ODBC接続)	○	○
	Linux	Symfoware Server	◎	○
		Oracle	○	○
		DB2	○	○
		PostgreSQL	○	○
		その他のデータベース(ODBC接続)	○	○
	相手側システムのデータベースから抽出	Windows	Symfoware Server	○
Oracle			○	○
SQL Server			○	○
DB2			○	○
SQL Azure			○	○
その他のデータベース(ODBC接続)			○	○
Solaris		Symfoware Server	○	○
		Oracle	○	○
		DB2	○	○

	Information Integratorサーバのプラットフォーム	データベース種別	機能	
			全件抽出	条件抽出
		その他のデータベース(ODBC接続)	○	○
	Linux	Symfoware Server	○	○
		Oracle	○	○
		DB2	○	○
		PostgreSQL	○	○
		その他のデータベース(ODBC接続)	○	○

◎:抽出方法として、アンローダ、または動的SQLを選択できます。(アンローダを選択した場合でも、自動的に動的SQLとなる場合があります。詳細は、“8.2.11 留意事項”を参照してください。

○:抽出方法は、動的SQLになります。

×:利用不可

## DB格納

表A.3 データベースごとの利用可能機能一覧(DB格納の場合)

	Information Integratorサーバのプラットフォーム	データベース種別	機能			
			データ置換	データ追加	データ更新	データ追加/更新
自システムのデータベースに格納	Windows	Symfoware Server	◎	○	○	○
		Oracle	◎	◎	○	○
		SQL Server	◎	◎	○	○
		DB2	○	○	○	○
		その他のデータベース(ODBC接続)	○	○	○	○
	Solaris	Symfoware Server	◎	○	○	○
		Oracle	◎	◎	○	○
		DB2	○	○	○	○
		その他のデータベース(ODBC接続)	○	○	○	○
	Linux	Symfoware Server	◎	○	○	○
		Oracle	◎	◎	○	○
		DB2	○	○	○	○
		PostgreSQL	○	○	○	○
		その他のデータベース(ODBC接続)	○	○	○	○
	相手側システムのデータベースに格納	Windows	Symfoware Server	○	○	○
Oracle			◎	◎	○	○
SQL Server			○	○	○	○
DB2			○	○	○	○

	Information Integratorサーバのプラットフォーム	データベース種別	機能			
			データ置換	データ追加	データ更新	データ追加/更新
		SQL Azure	○	○	○	○
		その他のデータベース (ODBC接続)	○	○	○	○
	Solaris	Symfoware Server	○	○	○	○
		Oracle	◎	◎	○	○
		DB2	○	○	○	○
		その他のデータベース (ODBC接続)	○	○	○	○
	Linux	Symfoware Server	○	○	○	○
		Oracle	◎	◎	○	○
		DB2	○	○	○	○
		PostgreSQL	○	○	○	○
その他のデータベース (ODBC接続)		○	○	○	○	

◎:格納方法として、ローダ、または動的SQLを選択できます。

○:格納方法は、動的SQLになります。

## A.3 オブジェクトごとの利用可能機能一覧

オブジェクトごとの抽出・格納で利用できる機能を、以下の表に示します。

### オブジェクト抽出

表A.4 オブジェクトごとの利用可能機能一覧(オブジェクト抽出の場合)

	オブジェクト種別	機能	
		全件抽出	条件抽出
相手側システムのオブジェクト抽出	Salesforce	○	○

○:抽出可

×:利用不可

### オブジェクト格納

表A.5 オブジェクトごとの利用可能機能一覧(オブジェクト格納の場合)

	オブジェクト種別	機能			
		データ置換	データ追加	データ更新	データ追加/更新
相手側システムのオブジェクトに格納	Salesforce	×	○	○	○

○:利用可

×:利用不可

## 付録B Information Integratorで扱えるデータ形式について

ここでは、Information Integratorで扱えるデータソースについて説明します。

### B.1 収集できるデータ形式

Information Integratorで収集できるデータ形式を以下に示します。

- ファイル形式
    - CSVファイル
    - バイナリファイル
    - XMLファイル
- 各ファイル形式のフォーマットについては、“付録D フォーマット形式”を参照してください。
- データベース形式
    - Symfoware Server
    - Oracle
    - SQL Server/SQL Azure
    - DB2
    - PostgreSQL
    - その他のデータベース(ODBC接続)
  - オブジェクト形式
    - Salesforceオブジェクト
  - Information Storage形式
  - サービスバス連携形式(Service Integrator形式)
  - BLOB形式

#### データ収集後のデータ形式

データ収集後に利用可能なデータ形式は、以下のとおりです。

表B.1 データ収集後に利用可能なデータ形式

収集種別		収集後のデータ形式		
		バイナリファイル	CSVファイル	XMLファイル
ファイル受信	バイナリ転送	○	注1	注1
	テキスト転送	×	○	○
ファイル入力		○	○	○
DB抽出	全件抽出	○	注2	×
	条件抽出	○	○	×
オブジェクト抽出	全件抽出	×	○	×
	条件抽出	×	○	×
IIS抽出	全件抽出	×	○	○
	条件抽出	×	○	○
サービスバス連携形式の入力		○	○	○
BLOB収集		○	○	○

○:可、×:不可

注1:改行コードが、自システムと同じ場合は可

注2:SQL\_LONGVARBINARY型以外は可

## 利用可能な改行コード

Information Integratorのデータ収集で利用可能な改行コードを以下の表に示します。

表B.2 データ収集で利用可能な改行コード

システム	利用可能な改行コード
Windows	CR+LF
UNIX	LF

## 収集元システムへのアクセス権

収集可能なファイル(CSVファイル、バイナリファイル、XMLファイル)のアクセス権は、以下のとおりに設定してください。

- ・ 収集元システムがWindowsの場合  
収集するファイルには、Information Integratorの特権ユーザーに対して、読み取り権が設定されている必要があります。
- ・ 収集元システムがUNIXの場合  
収集するファイルには、Information Integratorの特権ユーザーに対して、参照権が設定されている必要があります。

## B.1.1 Information Integratorで収集できるCSVファイル形式

Information Integratorで収集できるCSVファイルの形式は、以下のとおりです。

### 利用可能な形式

表B.3 収集できるCSVファイルの形式

項目	利用可能な範囲	備考
区切り文字	カンマ、半角空白、タブ	
囲み文字	なし、ダブルクォーテーション	
囲み文字をデータとして扱う場合	囲み文字をエスケープ文字とします。 例)"aaa","b""bb","ccc"	
区切り文字、囲み文字、改行コードをデータ内に含む場合	囲み文字で括る必要があります。	
項目数とデータの数不一致の場合	エラー(強制終了)	
空白のみのデータの場合	空白データとして扱います。	
1レコード	改行コードまで(囲み文字で括られている改行を除く)	
1レコードの制限	32000バイト	以下の制限があります。 ・ データチェック機能を利用する場合: 64000 バイト +23M バイト (SQL_LONGVARBINARY型のデータ) ・ 配付するデータ形式がバイナリファイルの場合: 32000 バイト +23M バイト (SQL_LONGVARBINARY型のデータ)

項目	利用可能な範囲	備考
		<ul style="list-style-type: none"> <li>配付先がDBMSの場合: 32000 バイト +31M バイト (SQL_LONGVARBINARY型のデータ)</li> </ul> なお、データ変換定義を利用しない場合、上記のSQL_LONGVARBINARY型のデータは、2Gバイトまで可能です。
1項目の制限	1レコードの制限に準拠	
項目名の文字数制限	128バイト	
項目数の制限	1024項目	
複数区切り文字の指定	指定可能 (連続した区切り文字を1つの区切り文字として扱います。)	
区切り文字前後の空白指定	不可	囲み文字を指定しなければ可能です。
最大ファイルサイズ	2Gバイト以上	

• データ形式について

- 1行分のデータは、列に対応するデータ項目の集まりで構成します。1つの列のデータ型に対応する各データ項目の間は、区切り文字(省略値はカンマ)で区切ります。

データ項目には、文字項目と数値項目があり、文字項目は囲み文字(ダブルクォーテーション(")のみサポート)で文字列を囲み、数値項目は、数値定数の文字列をそのまま指定します。また、1行分のデータの終了は、改行記号を指定します。

- データ項目にNULL値を指定する場合は、区切り文字の間に何も指定しない形で指定します。また、データ項目が1行の最後の列である場合は、直前のデータ項目との区切り文字を指定した後に改行記号が続くことになります。なお、NULL値のデータ項目は、NULL値識別フィールドを持ったバイナリファイル形式を使用する場合のみ扱うことが可能です。

— 空行の扱い

改行記号のみが存在する行は、すべての列にNULLが指定されたものとみなします。

— 0行ファイルの扱い

ファイルに行が存在しない場合は、0件のデータとして扱います。

- 項目数の最大は1024です。また、項目名の文字列長は半角36文字以下、全角18文字以下です。

- 項目名は、制御文字および半角空白を指定することはできません。

- 項目数とCSVファイル内のデータ数が一致しない場合、以下の動作となります。

- 項目数 > データ数・・・正常に動作します。不足項目は、NULL扱いで処理を継続します。
- 項目数 < データ数・・・エラーとなります。

• 区切り文字について

- 区切り文字に半角空白を指定した場合、NULL値を含めることはできません。

- 半角空白およびタブ以外の区切り文字を指定した場合、区切り文字間および囲み文字の外側と区切り文字の間にある半角空白およびタブは扱えません。

- 半角空白以外の区切り文字が連続する場合、それぞれの区切り文字の間にNULL値のデータが1個あるものとして扱います。

- あるデータ項目以降がすべてNULL値であっても、最後の項目まで区切り文字を指定する必要があります。

- 文字属性以外の項目データに区切り文字が存在する場合、囲み文字で囲むことはできません。

• 囲み文字について

- 囲み文字は、ダブルクォーテーションのみサポートします。

- 一 囲み文字で囲まれた文字列データ内に、囲み文字と同じ文字を使用する場合連続する囲み文字を2つ指定することにより、1つの文字データと見なします。

例: 囲み文字にダブルクォーテーションを指定した場合

- テキストの指定 : ~ , "A"BC" , ~
- データの値 : A"BC

連続2つの囲み文字の最初の文字(エスケープコード)は、文字数として計算しません。

- 一 囲み文字で囲まれた文字列データ内に、改行文字「¥n」と空文字「¥0」以外の制御文字が指定された場合、これらはデータとして扱います。
- 一 区切り文字を含む文字列データは囲み文字で囲む必要があります。

- 改行記号の扱いについて

WindowsとUNIXでは、改行の認識が異なります。

システム	出力時に設定する改行	読み込み可能な改行
Windows	0x0d0a	0x0d0a, 0x0a
UNIX	0x0a	0x0d0a, 0x0a

- 内部10進数について

有効な桁数が偶数の場合、先頭4ビットに設定されている値は無視します。

## 利用可能な文字コード

Information Integratorのデータ収集で利用可能な文字コードは、以下のとおりです。データチェック機能の利用有無により、利用可能な文字コードは異なります。

表B.4 利用可能な文字コード(データチェック機能を利用する場合)

利用可能な文字コード	
Shift-JIS	
Unicode	UTF8
	UTF8-4(注1)
	CFSD(注1)
EUC	U90(注1)
	S90(注2)
JEF-EBCDIC	カナ
	ASCII

注1) 以下の形式の場合、利用できません。

- 伝票形式
- 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

いずれもCSVファイルの場合は、プラグイン機能により「ifcnvdat」コマンドを実行し、利用可能な文字コード(Shift-JISなど)にコード変換することで利用することができます。

注2) Windowsの場合、かつ、以下の形式の場合は、利用できません。

- 伝票形式
- 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

いずれもCSVファイルの場合は、プラグイン機能により「ifcnvdat」コマンドを実行し、利用可能な文字コード(Shift-JISなど)にコード変換することで利用することができます。

表B.5 利用可能な文字コード(データチェック機能を利用しない場合)

利用可能な文字コード	
Shift-JIS	
Unicode	UCS2(注1)
	UTF8
	UTF8-4(注1)
	UCS2(MS)(注1)
	UTF16(注1)
	UTF16(MS)(注1)
EUC	U90(注1)
	S90(注2)
	HP(注1)
JEF-EBCDIC	カナ
	ASCII
IBM日本語文字セット	カナ
	英小文字

注1)以下の形式の場合、利用できません。

- ・ 伝票形式
- ・ 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

いずれもCSVファイルの場合は、プラグイン機能により「ifcnvdat」コマンドを実行し、利用可能な文字コード(Shift-JISなど)にコード変換することで利用することができます。

注2) Windowsの場合、かつ、以下の形式の場合は、利用できません。

- ・ 伝票形式
- ・ 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

いずれもCSVファイルの場合は、プラグイン機能により「ifcnvdat」コマンドを実行し、利用可能な文字コード(Shift-JISなど)にコード変換することで利用することができます。

## B.1.2 Information Integratorで収集できるバイナリファイル形式

Information Integratorで収集できるバイナリファイル形式は、以下のとおりです。

### 利用可能な形式

表B.6 収集できるバイナリファイルの形式

項目	利用可能な範囲	備考
1レコード	データ構造定義の「項目名」と「全体桁」に従います。	レコード終端に改行コードなどの制御文字は指定できません。
1レコードの制限	32000バイト+23Mバイト (SQL_LONGVARIABLE型 のデータ)	データ変換定義を利用しない場合、 SQL_LONGVARIABLE型のデータは2Gバイトまで可能です。
1項目の制限	1レコードの制限に準拠	
項目数の制限	1024項目	
最大ファイルサイズ	2Gバイト以上	



- バイナリファイルにおけるNULL値識別子フィールドの値について  
出力および参照可能な値は以下のとおりです。

データの値	出力時の設定値	読み込み可能な設定値
NULL値	0xFFFF	0xFFFF, 0x00FF
通常値	0x0000	0x0000

- Information Integratorで出力する数値のバイトオーダーは、動作するシステムに依存します。
  - Windowsの場合:後進法(little-endian)で出力します。
  - UNIXの場合:前進法(big-endian)で出力します。

## 利用可能な文字コード

Information Integratorのデータ収集で利用可能な文字コードを以下の表に示します。

表B.7 利用可能な文字コード

利用可能な文字コード	
Shift-JIS	
Unicode	UTF8
	UTF8-4(注1)
	CFSD(注1)
EUC	U90(注1)
	S90(注2)
JEF-EBCDIC	カナ
	ASCII

注1)以下の形式の場合、利用できません。

- 伝票形式
- 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

注2) Windowsの場合、かつ、以下の形式の場合は、利用できません。

- 伝票形式
- 一般(繰り返しあり)で、「項目繰り返し(可変)」が設定されている場合

## B.1.3 Information Integratorで収集できるXMLファイル形式

Information Integratorで収集できるXMLファイルの形式は、以下のとおりです。



混在フォーマットは利用できません。

### 利用可能な形式

表B.8 収集できるXMLファイルの形式

項目	利用可能な範囲	備考
XML文書の形式	「Extensible Markup Language(XML) 1.0」、 「XML Schema 1.0」、「XML名前空間」	XML規約で許されていないため、以下の文字は使用できません。 ・ 半角カタカナ

項目	利用可能な範囲	備考
	(Namespace in XML)」に準拠している整形式(well-formed)XML文書	<ul style="list-style-type: none"> <li>全角空白</li> <li>全角数字</li> <li>半角数字で始まるもの</li> <li>全角カンマ</li> <li>半角カンマ</li> </ul>
タグの長さ	ルートタグ名250バイト その他のタグ名36バイト	
空白タグ	以下の形式のものが扱えます。 <ul style="list-style-type: none"> <li>&lt;A&gt;&lt;/A&gt;</li> <li>&lt;A/&gt;</li> </ul>	

## 注意

次のXML文書は扱えません。

- タグの順序が不定なXML文書
- 文字参照を使用したXML文書
- 複数レコードが記載されたXML文書
- 再帰的な構造をもつXML文書
- 子要素を持たないタグと持つタグが同じタグ名であるXML文書
- 構成の異なる複数の親要素が同じタグ名を持つXML文書
- 任意の階層の中でタグ名が一意でない定義になるXML文書

## 利用可能な文字コード

Information Integratorのデータ収集で利用可能な文字コードは、「UTF8」です。

## 利用可能なXML Schema属性(データ型)と制約ファセット

XML Schema属性		サポートする制約ファセット
ビルトインデータ型	内容	
string	文字列	length(注)
boolean	ブール値	なし
decimal	10進数	totalDigits fractionDigits
float	単精度32ビット浮動小数	totalDigits fractionDigits
double	倍精度64ビット浮動小数	totalDigits fractionDigits
duration	期間	なし
dateTime	日時	なし
time	時刻	なし

XML Schema属性		サポートする制約ファセット
ビルトインデータ型	内容	
date	日付	なし
gYearMonth	年と月	なし
gYear	年	なし
gMonthDay	年の中の月日	なし
gDay	月の中の日	なし
gMonth	月	なし
hexBinary	16進表記のバイナリデータ	なし
base64Binary	Base64表記のバイナリデータ	なし
anyURI	URI参照値	なし
QName	名前空間で修飾されたXMLの名前	なし
NOTATION	XML 1.0のNOTATION属性	なし
normalizedString	正規化された文字列	length(注)
token	トークン化された文字列	length(注)
language	XML 1.0の言語識別子	なし
NMTOKEN	XML 1.0のNMTOKENS属性	なし
Name	XML 1.0の名前	なし
NCName	XML 1.0の名前からコロンを除いた名前	なし
ID	XML 1.0のID属性	なし
IDREF	XML 1.0のIDREF属性	なし
IDREFS	XML 1.0のIDREFS属性	なし
ENTITY	XML 1.0のENTITY属性	なし
ENTITIES	XML 1.0のENTITIES属性	なし
integer	整数	totalDigits
nonPositiveInteger	非正の整数	totalDigits
negativeInteger	負の整数	totalDigits
long	8バイト整数	totalDigits
int	4バイト整数	totalDigits
short	2バイト整数	totalDigits
byte	1バイト整数	totalDigits
nonNegativeInteger	非負の整数	totalDigits
unsignedLong	非負の8バイト整数	totalDigits
unsignedInt	非負の4バイト整数	totalDigits
unsignedShort	非負の2バイト整数	totalDigits
unsignedByte	非負の1バイト整数	totalDigits
positiveInteger	正の整数	totalDigits

注)属性がSQL\_CHAR、SQL\_NCHARの場合のみ

## B.1.4 Information Integratorで収集できるデータベース形式

Information Integratorで収集できるデータベースの形式を以下に示します。

### 抽出可能な文字コード

表B.9 Information Integratorで利用可能なSymfoware Serverの文字コード

文字コード		Symfoware Serverの文字コード							
		Windows		Solaris			Linux		
		Shift-JIS	Unicode	Shift-JIS	EUC	Unicode	Shift-JIS	EUC	Unicode
Shift-JIS		○		○					
Unicode	UCS2								
	UTF8								
	UTF8-4								
	UCS2(MS)								
	confused		○			○			○
	confused(MS)								
	UTF16								
EUC	U90								
	S90				○				
	HP								
JEF- EBCDIC	カナ(領域重視)								
	ASCII(領域重視)								
	カナ(字形重視)								
	ASCII(字形重視)								
IBM日本語 文字セット	カナ								
	英小文字								

○:利用可能

表B.10 Information Integratorで利用可能なOracleの文字コード

文字コード		Oracleの文字コード							
		Windows		Solaris			Linux		
		Shift-JIS	Unicode	Shift-JIS	EUC	Unicode	Shift-JIS	EUC	Unicode
Shift-JIS		○(注1)		○(注1)			○(注1)		
Unicode	UCS2								
	UFT8								
	UTF8-4		○(注3)			○(注3)			○(注3)
	UCS2(MS)								
	UTF16								
	UTF16(MS)								
EUC	U90								
	S90				○ (注2)			○ (注2)	

文字コード		Oracleの文字コード							
		Windows		Solaris			Linux		
		Shift-JIS	Unicode	Shift-JIS	EUC	Unicode	Shift-JIS	EUC	Unicode
	HP								
JEF- EBCDIC	カナ(領域重視)								
	ASCII(領域重視)								
	カナ(字形重視)								
	ASCII(字形重視)								
IBM日本語 文字セット	カナ								
	英小文字								

○:利用可能

注1)文字コードとして以下の3つのどれかを指定してください。

- JA16SJIS
- JA16SJISTILDE
- JA16SJISYEN

注2)文字コードとして以下の3つのどれかを指定してください。

- JA16EUC
- JA16EUCTILDE
- JA16EUCYEN

注3)文字コードとして以下を指定してください。

- AL32UTF8

表B.11 Information Integratorで利用可能なSQL Serverの文字コード

文字コード		SQL Serverの文字コード	
		Windows	
		Shift-JIS	Unicode
Shift-JIS		○	
Unicode	UCS2		
	UFT8		
	UTF8-4		
	UCS2(MS)		
	UTF16		
	UTF16(MS)		
EUC	U90		
	S90		
	HP		
JEF- EBCDIC	カナ(領域重視)		
	ASCII(領域重視)		
	カナ(字形重視)		
	ASCII(字形重視)		

文字コード		SQL Serverの文字コード	
		Windows	
		Shift-JIS	Unicode
IBM日本語 文字セット	カナ		
	英小文字		

○:利用可能

表B.12 Information Integratorで利用可能なDB2の文字コード

文字コード		DB2の文字コード							
		Windows		Solaris			Linux		
		Shift-JIS	Unicode	Shift-JIS	EUC	Unicode	Shift-JIS	EUC	Unicode
Shift-JIS		○							
Unicode	UCS2								
	UTF8								
	UTF8-4							○	
	UCS2(MS)								
	confused								
	confused(MS)								
	UTF16								
	UTF16(MS)								
EUC	U90								
	S90				○				
	HP								
JEF- EBCDIC	カナ(領域重視)								
	ASCII(領域重視)								
	カナ(字形重視)								
	ASCII(字形重視)								
IBM日本語 文字セット	カナ								
	英小文字								

○:利用可能

表B.13 Information Integratorで利用可能なPostgreSQLの文字コード

文字コード		PostgreSQLの文字コード		
		Linux		
		Shift-JIS	EUC	Unicode
Shift-JIS				
Unicode	UCS2			
	UTF8			
	UTF8-4			○
	UCS2(MS)			
	confused			

文字コード		PostgreSQLの文字コード		
		Linux		
		Shift-JIS	EUC	Unicode
	confused(MS)			
	UTF16			
	UTF16(MS)			
EUC	U90			
	S90			
	HP			
JEF- EBCDIC	カナ(領域重視)			
	ASCII(領域重視)			
	カナ(字形重視)			
	ASCII(字形重視)			
IBM日本語 文字セット	カナ			
	英小文字			

○:利用可能

## 抽出可能なデータ型

Information Integratorが抽出可能な、各データベースのデータ型、およびInformation Integratorと各データベースのデータ型の対応を以下に示します。



### 注意

- DB2、PostgreSQLおよびその他のデータベース(ODBC接続)では、データの抽出実行時には、収集元システムのデータベースの各項目のデータ型と、格納するデータボックスの各項目のデータ型の対応はチェックしません。事前に、データ型の対応を確認してください。
- 列数は、最大1024個です。制限を超える場合は、エラーとなります。

- Symfoware Serverの場合

Symfoware Serverの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.14 Symfoware Serverの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

Symfoware Serverのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
CHAR(n) n:1~32000	CHAR(n) n:1~32000	SQL_CHAR(m) m:1~32000	m=nの場合に抽出可能。
VARCHAR(n) n:1~32000	VARCHAR(n) n:1~32000	SQL_VARCHAR(m) m:1~32000	m=nの場合に抽出可能。
NCHAR(n) n:1~16000	NCHAR(n) n:1~16000	SQL_NCHAR(m) m:1~16000	m=nの場合に抽出可能。
NCHAR VARYING(n) n:1~16000	NCHAR VARYING(n) n:1~16000	SQL_VARNCHAR(m) m:1~16000	m=nの場合に抽出可能。

Symfoware Serverのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p2,q2) p2:1~18 q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
DECIMAL(p1,q1) p1:1~18 q1:0~p1	DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p2,q2) p2:1~18 q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
INTEGER	INTEGER	SQL_INTEGER	
SMALLINT	SMALLINT	SQL_SMALLINT	
FLOAT(p) p:1~52	FLOAT(p) p:1~52	SQL_REAL または SQL_FLOAT	pが1~23の場合はSQL_REAL、pが24~52の場合はSQL_FLOATで抽出可能。
REAL	REAL	SQL_REAL	
DOUBLE PRECISION	DOUBLE PRECISION	SQL_FLOAT	
DATE	DATE	SQL_DATE	
TIME	TIME	SQL_TIME	
TIMESTAMP	TIMESTAMP	SQL_TIMESTAMP	
INTERVAL各種	×		
BLOB(nK) n:1~2097152		SQL_LONGVARBINAR Y(m) m:1024~2146435072	m=n * 1024 の場合に抽出可能。 CSV形式での抽出はできません。
BLOB(nM) n:1~2048		SQL_LONGVARBINAR Y(m) m:1048576~2146435072	m=n * 1024^2 の場合に抽出可能。 CSV形式での抽出はできません。
BLOB(nG) n:1~2		SQL_LONGVARBINAR Y(m) m:1073741824 または m:2146435072	n=1の場合m=1073741824、 n=2の場合m=2146435072で抽出可能。 CSV形式での抽出はできません。

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

- Oracleの場合

Oracleの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。



表B.15 Oracleの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

Oracleのデータ型	Information Integratorサ ポート範囲	Information Integratorで 対応するデータ型	備考
VARCHAR2(n) n:1~4000	VARCHAR2(n) n:1~4000	SQL_VARCHAR(m) m:1~4000	m=nの場合に抽出可能。
NVARCHAR2(n) n:1~2000	NVARCHAR2(n) n:1~2000	SQL_VARNCHAR(m) m:1~2000	m=nの場合に抽出可能。
NUMBER(p1,q1) p1:1~38 q1:-84~127	NUMBER(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p2,q2) p2:1~18 q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
FLOAT(b) b:1~126	×		
LONG	×		
DATE	DATE	SQL_TIMESTAMP	
BINARY_FLOAT	×		
BINARY_DOUBLE	×		
TIMESTAMP(fsp) fsp:0~9	×		
TIMESTAMP(fs) WITH TIME ZONE fs:0~9	×		
TIMESTAMP(fs) WITH LOCAL TIME ZONE fs:0~9	×		
INTERVAL YEAR(yp) TO MONTH	×		
INTERVAL DAY(dp) TO SECOND(fs) dp:0~9 fs:0~9	×		
RAW(n) n:1~2000	×		
LONG RAW	LONG RAW	SQL_LONGVARBINAR Y(m) m:1~2147483647	mにはデータの格納できる サイズを指定してください。 CSV形式での抽出はでき ません。
ROWID	×		
UROWID(n) n:1~4000	×		
CHAR(n)	CHAR(n)	SQL_CHAR(m)	m=nの場合に抽出可能。

Oracleのデータ型	Information Integratorサ ポート範囲	Information Integratorで 対応するデータ型	備考
n:1~2000	n:1~2000	m:1~2000	
NCHAR(n) n:1~1000	NCHAR(n) n:1~1000	SQL_NCHAR(m) m:1~1000	m=nの場合に抽出可能。
CLOB	×		
NCLOB	×		
BLOB	×		
BFILE	×		

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

• SQL Serverの場合

SQL Serverの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.16 SQL Serverの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

SQL Serverのデータ 型	Information Integratorサ ポート範囲	Information Integratorで対 応するデータ型	備考
bigint	×		
int	int	SQL_INTEGER	
smallint	smallint	SQL_SMALLINT	
tinyint	×		
numeric(p1,q1) p1:1~38 q1:0~p1	numeric(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p2,q2) p2:1~18、q2:0~p2	p2=p1,q2=q1の場合に 抽出可能。
decimal(p1,q1) p1:1~38 q1:0~p1	decimal(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p2,q2) p2:1~18、q2:0~p2	p2=p1,q2=q1の場合に 抽出可能。
bit	×		
money	×		
smallMoney	×		
float(n) n:1~53	float(n) n:1~53	SQL_REAL または SQL_FLOAT	pが1~24の場合は SQL_REAL、pが25~ 53の場合は SQL_FLOATで抽出可 能。
real	real	SQL_REAL	
date	×		
datetimeoffset	×		
datetime2	×		
smallDatetime	smallDatetime	SQL_TIMESTAMP	
datetime	datetime	SQL_TIMESTAMP	

SQL Serverのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
time	×		
char(n) n:1～8000	char(n) n:1～8000	SQL_CHAR(m) m:1～8000	m=nの場合に抽出可能。
varchar(n) n:1～8000,MAX	varchar(n) n:1～8000	SQL_VARCHAR(m) m:1～8000	m=nの場合に抽出可能。
text	×		
nchar(n) n:1～4000	×		
nvarchar(n) n:1～4000,MAX	×		
ntext	×		
binary(n) n:1～8000	×		
varbinary(n) n:1～8000,MAX	×		
image(n) n:1～2147483647	image(n) n:1～2147483647	SQL_LONGVARBINARY(m) m:1～2147483647	mにはデータの格納できるサイズを指定してください。 CSV形式での抽出はできません。
cursor	×		
timestamp	×		
hierarchyid	×		
uniqueidentifier	×		
sql_variant	×		
xml	×		
table	×		
geometry	×		
geography	×		
sysname	×		
FILESTREAM	×		

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

- DB2の場合

DB2の場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.17 DB2の場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

DB2のデータ型	Information Integratorサ ポート範囲	Information Integratorで対 応するデータ型	備考
CHAR(n) n:1~254	CHAR(n) n:1~254	SQL_CHAR(m) m:1~254	m=nの場合に抽出可能。
VARCHAR(n) n:1~32672	VARCHAR(n) n:1~32000	SQL_VARCHAR(m) m:1~32000	m=nの場合に抽出可能。
NUMERIC(p1,q1) p1:1~31 q1:0~p1	NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p2,q2) p2:1~18 q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
DECIMAL(p1,q1) p1:1~31 q1:0~p1	DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p2,q2) p2:1~18 q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
DECFLOAT(n) n:16または34	×		
INTEGER	INTEGER	SQL_INTEGER	
SMALLINT	SMALLINT	SQL_SMALLINT	
FLOAT(p) p:1~53	×		
REAL	×		
BIGINT	×		
DOUBLE	×		
DOUBLE PRECISION	×		
DATE	DATE	SQL_DATE	
TIME	TIME	SQL_TIME	
TIMESTAMP	TIMESTAMP	SQL_TIMESTAMP	
LONG VARCHAR	×		
CLOB(n) n:1~2147483647	×		
CLOB(nK) n:1~2097152	×		
CLOB(nM) n:1~2048	×		
CLOB(nG) n:1~2	×		
GRAPHIC(n) n:1~127	×		
VARGRAPHIC(n) n:1~16336	×		

DB2のデータ型	Information Integratorサ ポート範囲	Information Integratorで対 応するデータ型	備考
LONG VARGRAPHIC	×		
DBCLOB(n) n:1~1073741823	×		
DBCLOB(nK) n:1~1048576	×		
DBCLOB(nM) n:1~1024	×		
DBCLOB(nG) n:1	×		
BLOB(n) n:1~2147483647	×		
BLOB(nK) n:1~2097152	×		
BLOB(nM) n:1~2048	×		
BLOB(nG) n:1~2	×		
XML	×		

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

• PostgreSQLの場合

PostgreSQLの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.18 PostgreSQLの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

PostgreSQLのデータ型	Information Integratorサ ポート範囲	Information Integratorで対 応するデータ型	備考
bigint	×		
bigserial	×		
bit	×		
bit varying(n) n:1~制限なし	×		
boolean	×		
box	×		
bytea	×		
character(n) n:1~制限なし	character(n) n:1~32000	SQL_CHAR(m) m:1~32000	m=nの場合に抽出可能。

PostgreSQLのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
character varying(n) n:1～制限なし	character varying(n) n:1～32000	SQL_VARCHAR(m) m:1～32000	m=nの場合に抽出可能。
cidr	×		
circle	×		
date	date	SQL_DATE	
double precision	×		
inet	×		
integer	integer	SQL_INTEGER	
interval(p) p:0～6	×		
line	×		
lseg	×		
macaddr	×		
money	×		
numeric [ (p, s) ] p:1～1000 s:0～p	numeric [ (p, s) ] p:1～18s:0～p	SQL_NUMERIC(p2,q2) p2:1～18 q2:0～p2 または SQL_DECIMAL(p2,q2) p2:1～18 q2:0～p2	p2=p1,q2=q1の場合に抽出可能。
oid	×		
path	×		
point	×		
polygon	×		
real	×		
smallint	smallint	SQL_SMALLINT	
serial	×		
text	×		
time [ (p) ] [ without time zone ] p:0～10	×		
time [ (p) ] with time zone p:0～10	×		
timestamp [ (p) ] [without time zone] p:0～6	×		
timestamp [ (p) ] with time zone p:0～6	×		

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

• SQL Azureの場合

SQL Azureの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.19 SQL Azureの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

SQL Azureのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
bigint	×		
int	int	SQL_INTEGER	
smallint	smallint	SQL_SMALLINT	
tinyint	×		
numeric(p1,q1) p1:1~38 q1:0~p1	numeric(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p2,q2) p2:1~18、q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
decimal(p1,q1) p1:1~38 q1:0~p1	decimal(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p2,q2) p2:1~18、q2:0~p2	p2=p1,q2=q1の場合に抽出可能。
bit	×		
money	×		
smallMoney	×		
float(n) n:1~53	float(n) n:1~53	SQL_REAL または SQL_FLOAT	pが1~24の場合はSQL_REAL、pが25~53の場合はSQL_FLOATで抽出可能。
real	real	SQL_REAL	
date	×		
datetimeoffset	×		
datetime2	×		
smallDatetime	smallDatetime	SQL_TIMESTAMP	
datetime	datetime	SQL_TIMESTAMP	
time	×		
char(n) n:1~4000	char(n) n:1~4000	SQL_CHAR(m) m:1~4000	m=nの場合に抽出可能。
varchar(n) n:1~4000,MAX	varchar(n) n:1~4000	SQL_VARCHAR(m) m:1~4000	m=nの場合に抽出可能。
text	×		
nchar(n) n:1~4000	×		
nvarchar(n)	×		

SQL Azureのデータ型	Information Integratorサポート範囲	Information Integratorで対応するデータ型	備考
n:1~4000,MAX			
ntext	×		
binary(n) n:1~8000	×		
varbinary(n) n:1~8000,MAX	×		
image(n) n:1~2147483647	image(n) n:1~2147483647	SQL_LONGVARBINAR Y(m) m:1~2147483647	mにはデータの格納できるサイズを指定してください。 CSV形式での抽出はできません。
cursor	×		
timestamp	×		
hierarchyid	×		
uniqueidentifier	×		
sql_variant	×		
xml	×		
table	×		
geometry	×		
geography	×		
sysname	×		
FILESTREAM	×		

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

## B.1.5 Information Integratorで収集できるオブジェクト形式

Information Integratorで収集できるオブジェクト形式を以下に示します。

### 抽出したデータのInformation Integratorでのデータ形式

- Salesforceオブジェクトの場合

Salesforceから抽出したデータは、UnicodeのUTF8、改行コードはCR+LFで扱います。また、抽出したCSVファイルの区切り文字は、「,」(カンマ)で扱います。

### 抽出可能なデータ型

- Salesforceオブジェクトの場合

Salesforceオブジェクトの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。



表B.20 Salesforceオブジェクトの場合の、抽出可能なデータ型、およびInformation Integratorのデータ型との対応

Salesforceオブジェクト形式のデータ型 (FieldType)	Information Integratorで対応するデータ型	備考
string	SQL_VARCHAR	文字列 SalesforceのbyteLengthをInformation Integratorの全桁数として扱います。
boolean	SQL_VARCHAR	ブール値(true/false) Information Integratorでは、「true」または「false」の文字列表現として扱います。
int	SQL_INTEGER(注1)	整数 Salesforceから抽出したデータには、ダブルクォーテーションが付与されます。
double	SQL_FLOAT(注1)	浮動小数点 Salesforceから抽出したデータには、ダブルクォーテーションが付与されます
date	SQL_DATE	日付 YYYY-MM-DD (例)2010-04-08 Salesforceから協定世界時(UTC)が抽出されます。
datetime	SQL_VARCHAR(注2)	日付／時間 YYYY-MM-DDTHH:MM:SS.nnnZ (例)2010-04-08T22:15:33.000Z Salesforceから協定世界時(UTC)が抽出されます。
base64	SQL_LONGVARBINARY	BASE64 CSV形式のデータの中では、値はBASE64形式で符号化されています。 メタ収集時、Information Integratorの全桁数に5242880(固定値)が設定されます。ただし、オブジェクト抽出時のデータの桁数は、データごとにそれぞれ異なります。
ID	SQL_CHAR	ID
reference	×(注3)	参照型
currency	SQL_FLOAT(注1)	通貨(¥10,000など) Salesforceから抽出したデータには、ダブルクォーテーションが付与されます
textarea	SQL_VARCHAR	テキストエリア
percent	SQL_FLOAT(注1)	パーセント(80%など) Salesforceから抽出したデータには、ダブルクォーテーションが付与されます
phone	SQL_VARCHAR	電話(000-111-2222など) メタ収集時、Salesforceが保持している(SalesforceのAPIで取得できる)桁数がInformation Integratorの全桁数として設定されます。

Salesforceオブジェクト形式のデータ型 (FieldType)	Information Integratorで対応するデータ型	備考
url	SQL_VARCHAR	URL メタ収集時、Salesforceが保持している(SalesforceのAPIで取得できる)桁数がInformation Integratorの全桁数として設定されます。
email	SQL_VARCHAR	電子メール メタ収集時、Salesforceが保持している(SalesforceのAPIで取得できる)桁数がInformation Integratorの全桁数として設定されます。
combobox	SQL_VARCHAR	コンボボックス GUI上ではリストから選択ができ、文字列入力もできる
picklist	SQL_VARCHAR	選択リスト メタ収集時、Salesforceが保持している(SalesforceのAPIで取得できる)桁数がInformation Integratorの全桁数として設定されます。
multipicklist	SQL_VARCHAR	選択リスト(複数選択) メタ収集時、Salesforceが保持している(SalesforceのAPIで取得できる)桁数がInformation Integratorの全桁数として設定されます。
anyType	SQL_VARCHAR	任意のデータ型 Salesforceでは標準オブジェクトの標準項目で以下のデータ属性として使用されています。 <ul style="list-style-type: none"> <li>• string</li> <li>• picklist</li> <li>• boolean</li> <li>• int</li> <li>• double</li> <li>• percent</li> <li>• ID</li> <li>• date</li> <li>• datetime</li> <li>• url</li> <li>• email</li> </ul>

×:利用不可

Information Integratorで対応するデータ型のデータ例は、“[B.3 Information Integratorで対応するデータ型のデータ例](#)”を参照してください。

注1)

Salesforceから抽出したデータには、ダブルクォーテーションが付与されます。そのため、収集データ(データチェック時の入力側のデータ)に対してデータチェックを行うことはできません。

注2)

日時のデータ「YYYY-MM-DDTHH:MM:SS.nnnZ(年-月-日T時間:分:秒.ミリ秒 TおよびZは区切り文字)」を抽出した場合、データは、文字として扱われます(SQL\_VARCHAR)。

日時データとして扱いたい場合は、Information Integratorが扱える形式「YYYY-MM-DD HH:MM:SS(年-月-日 時間:分:秒)」にデータを変換する必要があります。区切り文字およびミリ秒のデータを削除するアプリケーションを作成し、処理プラグインを組み込んでください。

注3)

メタ収集時にInformation Integratorで利用できない形式のデータ型が含まれていた場合、利用できない形式のメタを無効として、抽出可能なメタデータだけ収集します。メタ収集が完了した後、利用できないSalesforceオブジェクト形式のデータ型が含まれていたことを示す警告メッセージが出力されます。



### Information Integratorで抽出可能なSalesforceオブジェクトのデータ項目(Field)

Information Integratorで抽出可能なSalesforceオブジェクトのデータ項目(Field)は、以下のとおりです。

- byteLength
- digits
- label
- length
- nillable
- precision
- scale
- type
- unique

## B.1.6 Information Integratorで収集できるInformation Storage形式

Information Integratorで収集できるInformation Storage形式を以下に示します。

### 利用可能な文字コード

Information Integratorのデータ収集で利用可能な文字コードは、以下のとおりです。

利用可能な文字コード	
Unicode	UTF8

### 利用可能な文字

Information Integratorで収集するデータの項目名、タグ名は、以下の文字を利用できます。

- すべてのマルチバイト文字
- 以下のシングルバイト文字

```
! 0 1 2 3 4 5 6 7 8 9 : @ A B C
D E F G H I J K L M N O P Q R S T
U V W X Y Z _ ` a b c d e f g h i
```

j k l m n o p q r s t u v w x y z

## 利用可能なデータサイズ

収集可能なCSVファイルの1レコード長および収集可能なXMLファイルの1文書のサイズは、最大32,000バイトまでです。

## B.1.7 Information Integratorで収集できるBLOB形式

Information Integratorでは、BLOBに格納されているすべてのデータを収集することができます。

## B.2 配付できるデータ形式

Information Integratorで配付できるデータ形式を以下に示します。

- ファイル形式
  - CSVファイル
  - バイナリファイル
  - XMLファイル
- データベース形式
  - Symfoware Server
  - Oracle
  - SQL Server / SQL Azure
  - DB2
  - PostgreSQL
  - その他のデータベース(ODBC接続)
- オブジェクト形式
  - Salesforceオブジェクト
- 帳票データ
  - シングルフォーム
  - マルチフォーム
- Information Storage形式
- サービスバス連携形式(Service Integrator形式)
- BLOB形式

### データ格納前のデータ形式

データ格納前に利用可能なデータ形式は、以下のとおりです。

表B.21 データ格納前に利用可能なデータ形式

格納種別		格納前のデータ形式		
		バイナリファイル	CSVファイル	XMLファイル
ファイル送信	バイナリ転送	○	注1	注1
	テキスト転送	×	○	○
ファイル出力		○	○	○
DB格納	全データ置換	○	○	×
	追加	○	○	×

格納種別		格納前のデータ形式		
		バイナリファイル	CSVファイル	XMLファイル
	更新	○	○	×
	追加更新	○	○	×
オブジェクト格納		×	○	×
帳票出力		×	○	×
IIS格納		×	○	○
サービスバス連携形式の出力		○	○	○
BLOB格納		○	○	○

○:利用可、×:利用不可

注1:改行コードが、自システムと同じ場合は可

## 利用可能な改行コード

Information Integratorのデータ配付で利用可能な改行コードを以下の表に示します。

表B.22 利用可能な改行コード

システム	利用可能な改行コード
Windows	CR+LF
UNIX	LF

## 配付先システムへのアクセス権

配付されるファイル(CSVファイル、バイナリファイル)のアクセス権は、以下のとおりに設定されます。

- 配付先システムがWindowsの場合  
配付するファイルのアクセス権は、Information Integratorの特権ユーザーに対して、読み書き権が設定されます。
- 配付先システムがUNIXの場合  
配付するファイルのアクセス権は、Information Integratorの特権ユーザー、および所属グループに対して、参照および書き込み権が設定されます。

## B.2.1 Information Integratorで配付できるCSVファイル形式

Information Integratorで配付できるCSVファイル形式は、収集できるCSVファイル形式と同じです。

利用可能な形式については、“[B.1.1 Information Integratorで収集できるCSVファイル形式](#)”を参照してください。

## B.2.2 Information Integratorで配付できるバイナリファイル形式

Information Integratorで配付できるバイナリファイル形式は、収集できるバイナリファイル形式と同じです。

利用可能な形式については、“[B.1.2 Information Integratorで収集できるバイナリファイル形式](#)”を参照してください。

## B.2.3 Information Integratorで配付できるXMLファイル形式

Information Integratorで配付できるXMLファイルの形式は、以下のとおりです。

Information Integratorで配付できるXMLファイル形式は、収集できるXMLファイル形式と同じです。

利用可能な形式については、“[B.1.3 Information Integratorで収集できるXMLファイル形式](#)”を参照してください。

## B.2.4 Information Integratorで配付できるデータベース形式

Information Integratorで格納できるデータベースの形式を以下に示します。

### 格納可能な文字コード

抽出可能な文字コードと同様です。抽出可能な文字コードについては、“[B.1.4 Information Integratorで収集できるデータベース形式](#)”を参照してください。

### 格納可能なデータ型

Information Integratorが格納可能な、Information Integratorのデータ型と各データベースのデータ型の対応を以下に示します。

ここで記載している各データベースのデータ型の範囲は、各データベースのもので、Information Integratorで扱えるデータの範囲を超えて格納することはできません。テーブルを作成する場合には、対応に従って、各項目のデータ型が定義されます。



### 注意

- 列数は、最大1024個です。制限を超える場合は、エラーとなります。

- Symfoware Serverの場合

Symfoware Serverへの格納が可能なデータ型の対応を以下の表に示します。

表B.23 Symfoware Serverへの格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSymfoware Serverのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~32000	CHAR(m) m:1~32000	n <= m の場合に格納可能。
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~32000	VARCHAR(m) m:1~32000	n <= m の場合に格納可能。
SQL_NCHAR(n) n:1~16000	SQL_NCHAR(n) n:1~16000	NCHAR(m) m:1~16000	n <= m の場合に格納可能。  Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。  Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_VARNCHAR(n) n:1~16000	SQL_VARNCHAR(n) n:1~16000	NCHAR VARYING(m) m:1~16000	n <= m の場合に格納可能。  Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。  Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_INTEGER	SQL_INTEGER	INTEGER	

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSymfoware Serverのデータ型と範囲	備考
SQL_SMALLINT	SQL_SMALLINT	SMQLLINT	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	DECIMAL(p2,q2) p2:1~18 q2:0~p2	p1=p2、q1=q2の場合に格納可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p2,q2) p2:1~18 q2:0~p2	p1=p2、q1=q2の場合に格納可能。
SQL_REAL	SQL_REAL	REAL FLOAT(p) p:1~23	
SQL_FLOAT	SQL_FLOAT	DOUBLE PRECISION FLOAT(p) p:24~52	
SQL_DATE	SQL_DATE	DATE	
SQL_TIME	SQL_TIME	TIME	
SQL_TIMESTAMP	SQL_TIMESTAMP	TIMESTAMP	
SQL_INTERVALTIME	×		
SQL_LONGVARIABLE(n) n:1~2147483647	SQL_LONGVARIABLE(n) n:1~2146435072	BLOB(kK) k:1~2096128 BLOB(mM) m:1~2047 BLOB(gG) g:1~2	以下の場合に格納可能。 n <= k * 1024 n <= m * 1024^2 n <= g * 1024^3

×:利用不可

• Oracleの場合

Oracleへの格納が可能なデータ型の対応を以下の表に示します。

表B.24 Oracleへの格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するOracleのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~32000	CHAR(m) m:1~2000 VARCHAR2(m) m:1~4000 LONG	n <= m の場合に格納可能。

Information Integratorのデータ型	データベースに出力可能な範囲	対応するOracleのデータ型と範囲	備考
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~32000	VARCHAR2(m) m:1~4000 LONG	n <= m の場合に格納可能。
SQL_NCHAR(n) n:1~16000	SQL_NCHAR(n) n:1~16000	NCHAR(m) m:1~(バイト数換算で2000) NVARCHAR2(m) m:1~(バイト数換算で4000) LONG	n <= m の場合に格納可能。 (1文字のバイト数はキャラクタセットによって変化します) Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。 Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_VARNCHAR(n) n:1~16000	SQL_VARNCHAR(n) n:1~16000	VARCHAR2(m) m:1~(バイト数換算で4000) LONG	n <= m の場合に格納可能。 (1文字のバイト数はキャラクタセットによって変化します) Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。 Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_INTEGER	SQL_INTEGER	NUMBER(p,0) p:10~18	
SQL_SMALLINT	SQL_SMALLINT	NUMBER(p,0) p:5~18	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	NUMBER(p2,q2) p2:1~18 q2:0~p2	p1=p2、q1=q2の場合に格納可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMBER(p2,q2) p2:1~18 q2:0~p2	p1=p2、q1=q2の場合に格納可能。
SQL_REAL	SQL_REAL	NUMBER	
SQL_FLOAT	SQL_FLOAT	NUMBER	
SQL_DATE	SQL_DATE	DATE	



Information Integratorのデータ型	データベースに出力可能な範囲	対応するOracleのデータ型と範囲	備考
SQL_TIME	SQL_TIME	DATE	
SQL_TIMESTAMP	SQL_TIMESTAMP	DATE	
SQL_INTERVALTIME	×		
SQL_LONGVARIABLEY(n) n:1~2147483647	SQL_LONGVARIABLEY(n) n:1~2147483647	LONG RAW	

×:利用不可

• SQL Serverの場合

SQL Serverへの格納が可能なデータ型の対応を以下の表に示します。

表B.25 SQL Serverへの格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSQL Serverのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~32000	char(m) m:1~8000	n <= m の場合に格納可能。
		text	
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~32000	varchar(m) m:1~8000	n <= m の場合に格納可能。
		text	
SQL_NCHAR(n) n:1~16000	×		
SQL_VARNCHAR(n) n:1~16000	×		
SQL_INTEGER	SQL_INTEGER	int	
SQL_SMALLINT	SQL_SMALLINT	smallint	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	decimal(p2,q2) p2:1~18 q2:0~p2	p1=p2,q1=q2の場合に格納可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	numeric(p2,q2) p2:1~18 q2:0~p2	p1=p2,q1=q2の場合に格納可能。
SQL_REAL	SQL_REAL	real	
SQL_FLOAT	SQL_FLOAT	float(m) m:53	
SQL_DATE	×		
SQL_TIME	×		
SQL_TIMESTAMP	SQL_TIMESTAMP	datetime	

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSQL Serverのデータ型と範囲	備考
		smalldatetime	
SQL_INTERVALTIME	×		
SQL_LONGVARIABLE(n) n:1~2147483647	SQL_LONGVARIABLE(n) n:1~2147483647	image	

×:利用不可

・DB2の場合

DB2への格納が可能なデータ型の対応を以下の表に示します。

表B.26 DB2への格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するDB2のデータ型と範囲	備考
SQL_CHAR(m) m:1~254	SQL_CHAR(m) m:1~254	CHAR(n) n:1~254	m=nの場合に格納可能。
SQL_VARCHAR(m) m:1~32000	SQL_VARCHAR(m) m:1~32000	VARCHAR(n) n:1~32672	m=nの場合に格納可能。
SQL_NCHAR(n) n:1~16000	×		
SQL_VARNCHAR(n) n:1~16000	×		
SQL_INTEGER	SQL_INTEGER	INTEGER	
SQL_SMALLINT	SQL_SMALLINT	SMALLINT	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	DECIMAL(p2,q2) p2:1~31 q2:0~p1	p1=p2,q1=q2の場合に格納可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p2,q2) p2:1~31 q2:0~p1	
SQL_REAL	×		
SQL_FLOAT	×		
SQL_DATE	SQL_DATE	DATE	
SQL_TIME	SQL_TIME	TIME	
SQL_TIMESTAMP	SQL_TIMESTAMP	TIMESTAMP	
SQL_INTERVALTIME	×		
SQL_LONGVARIABLE(n) n:1~2147483647	×		

×:利用不可

- PostgreSQLの場合

PostgreSQLへの格納が可能なデータ型の対応を以下の表に示します。

表B.27 PostgreSQLへの格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するPostgreSQLのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~32000	Character(n) N:1~32000	
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~32000	Character varying(n) N:1~32000	
SQL_NCHAR(n) n:1~16000	×		
SQL_VARNCHAR(n) n:1~16000	×		
SQL_INTEGER	SQL_INTEGER	Integer	
SQL_SMALLINT	SQL_SMALLINT	Smallint	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p2,q2) P2:1~18 Q2:0~p2	
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	NUMERIC(p2,q2) P2:1~18 Q2:0~p1	
SQL_REAL	×		
SQL_FLOAT	×		
SQL_DATE	SQL_DATE	Date	
SQL_TIME	×		
SQL_TIMESTAMP	×		
SQL_INTERVALTIME	×		
SQL_LONGVARIABLE(n) n:1~2147483647	×		
SQL_BIT(n) N:1~128	×		
SQL_BINARY(n) N: 1~2147483647	×		

- SQL Azureの場合

SQL Azureへの格納が可能なデータ型の対応を以下の表に示します。

表B.28 SQL Azureへの格納が可能なデータ型

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSQL Azureのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~32000	char(m) m:1~4000	n <= m の場合に格納可能。

Information Integratorのデータ型	データベースに出力可能な範囲	対応するSQL Azureのデータ型と範囲	備考
		text	
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~32000	varchar(m) m:1~4000 text	n <= m の場合に格納可能。
SQL_NCHAR(n) n:1~16000	×		
SQL_VARNCHAR(n) n:1~16000	×		
SQL_INTEGER	SQL_INTEGER	int	
SQL_SMALLINT	SQL_SMALLINT	smallint	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	decimal(p2,q2) p2:1~18 q2:0~p2	p1=p2,q1=q2の場合に格納可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	numeric(p2,q2) p2:1~18 q2:0~p2	p1=p2,q1=q2の場合に格納可能。
SQL_REAL	SQL_REAL	real	
SQL_FLOAT	SQL_FLOAT	float(m) m:53	
SQL_DATE	×		
SQL_TIME	×		
SQL_TIMESTAMP	SQL_TIMESTAMP	datetime smalldatetime	
SQL_INTERVALTIME	×		
SQL_LONGVARBINARY(n) n:1~2147483647	SQL_LONGVARBINARY(n) n:1~2147483647	image	

×:利用不可

## B.2.5 Information Integratorで配付できるオブジェクト形式

Information Integratorで配付できるオブジェクト形式を以下に示します。

### 格納したデータのInformation Integratorでのデータ形式

- Salesforceオブジェクトの場合

Salesforceへ格納するデータは、UnicodeのUTF8、改行コードはCR+LFで扱います。また、格納するCSVファイルの区切り文字は、「,(カンマ)」で扱います。

その他の形式の区切り文字(空白、タブ)がデータに含まれる場合は、格納前にデータ変換・編集機能を利用して、あらかじめ区切り文字を「,(カンマ)」にしてください。その他の形式の区切り文字(空白、タブ)がデータに含まれたまま格納しようとすると、処理異常となります。

## 格納可能なデータ型

- Salesforceオブジェクトの場合

Salesforceオブジェクトの場合の、格納可能なデータ型、およびInformation Integratorのデータ型との対応を以下の表に示します。

表B.29 Salesforceオブジェクトの場合の、格納可能なデータ型、およびInformation Integratorのデータ型との対応

Information Integratorで対応するデータ型	Salesforceオブジェクト形式のデータ型(FieldType)	備考
SQL_CHAR	ID	格納することはできません。したがって、Salesforce側で保持する当該IDの値は変更されません。 データ更新の場合、レコード管理IDとして値は参照されます。
SQL_VARCHAR(注1)	string	
	boolean	以下のいずれかで格納可能 <ul style="list-style-type: none"> <li>• 「true」または「false」の文字列で格納</li> <li>• 「true(1)」、「false(0)」の、それぞれの括弧内の数値で格納</li> </ul>
	datetime	協定世界時(UTC)で格納してください。
	textarea	
	phone	Salesforceの仕様に合わせた形式の値を格納可
	url	Salesforceの仕様に合わせた形式の値を格納可
	email	Salesforceの仕様に合わせた形式の値を格納可
	combobox	
	picklist	Salesforceの選択リストに含まれていないデータも格納可。Salesforceの選択リストに含まれていないデータを格納すると、選択リストに追加されます。
	multipicklist	
anyType	Salesforceの仕様に合わせた形式の値を格納可	
SQL_INTEGER	int	配付時の最大値(最小値)は、Salesforceで定義されている桁数に従います。
SQL_FLOAT	double	配付時の最大値(最小値)は、Salesforceで定義されている桁数に従います。
	currency	配付時の最大値(最小値)は、Salesforceで定義されている桁数に従います。
	percent	配付時の最大値(最小値)は、Salesforceで定義されている桁数に従います。
SQL_DATE(注2)	date	協定世界時(UTC)で格納してください。
SQL_LONGVARIABLEY(注3)	base64	

注1)

Salesforceオブジェクト形式のデータ型「datetime」で格納するデータは、「YYYY-MM-DDTHH:MM:SS.nnnZ(年-月-日T時間:分:秒.ミリ秒 TおよびZは区切り文字)」の形式にする必要があります。

Information Integratorの日時データを格納する場合には、Information Integratorが扱える形式「YYYY-MM-DD HH:MM:SS(年-月-日 時間:分:秒)」から格納可能なデータ形式にデータを変換する必要があります。区切り文字およびミリ秒を追加するアプリケーションを作成し、処理プラグインを組み込んでください。

注2)

Salesforceオブジェクトを格納するとき、日付データは、「YYYY-MM-DD」の形式にしてください。

その他の形式で表現されている日付データをSalesforceに格納する場合は、日時の書式変換機能を利用して形式を「YYYY-MM-DD」の形式にそろえてください。日時の書式変換についての詳細は、「[1.4.3.17 日時の書式変換](#)」を参照してください。

注3)

Salesforceに格納するデータは、Base64で符号化してください。

その他の形式で表現されているデータの場合は、属性変換機能を利用して形式を変換してください。属性変換機能についての詳細は、「[1.4.3.4 属性変換](#)」を参照してください。

## Salesforceの外部IDに関する留意事項

データの追加/更新を行うときに利用するSalesforceの外部IDは、Salesforce側で必ず一意となっているキーを外部IDにしてください。

## B.2.6 Information Integratorで出力できる帳票データ

Information Integratorで出力できる帳票データの形式は、以下のとおりです。

### 利用可能な形式

帳票データは、Information IntegratorからInterstage List CreatorへCSVファイル形式で受け渡し、出力します。

CSVファイル形式については、「[B.2.1 Information Integratorで配付できるCSVファイル形式](#)」を参照してください。

あわせて、以下のデータ形式にしてください。

- マルチフォーム出力を行う場合、出力するすべての帳票のデータ形式を統一してください。
- マルチフォーム出力を行う場合、レコード内のデータが切り替わるポイントでデータを振り分ける場合(※)は、すべての帳票に対して振り分けを行います。配付データの中に、振り分けを行うデータと行わないデータが混在しないようにしてください。また、振り分けを行う場合は、すべての振り分けキー項目には同じ意味づけのデータ群にしてください。振り分けを行う項目の値によって、必ず事前に同じ方式のソートが実施されているデータにしてください。  
※)レコード内のデータが切り替わるポイントでデータを振り分ける場合は、プラグイン利用時に行うことができます。詳細は、後述の「[プラグインを利用してInterstage List Creatorにデータを渡す](#)」を参照してください。
- 帳票定義体のデータ定義にある入力データ形式に応じて、あらかじめ囲み文字を付与する/しないを設定してください。ただし、カンマ、半角空白、タブのいずれかを含むデータの場合は、囲み文字の付与が必須のため、帳票定義体の入力データ形式に「CSV形式」に設定してください。

帳票定義体の入力データ形式	格納するデータの囲み文字
CSV形式	文字型データのみ囲み文字を付与する
可変長	すべてのデータに囲み文字を付与しない

なお、囲み文字は、データ変換・編集機能で設定することができます。

- 配付可能なCSVファイルの1レコードの最大バイト数は、65535バイトです。

### Interstage List Creatorの帳票定義体の設定に関する留意事項

- 入力データ形式の選択では、「可変長」または「CSV形式」を選択してください。ただし、カンマ、半角空白、タブのいずれかを含むデータの場合は、必ず「CSV形式」を設定してください。

- 項目種別が数字で、かつ、小数部桁数が存在する場合、小数部桁数を17桁以内にしてください。18桁以上の場合、処理プロセスの実行時にエラーとなります。
- 矩形テキストの改行コードの指定は、無効です。
- レポート頭書き/RHの指定はしないでください。
- 区切り文字の指定は、カンマ、タブ、半角空白のいずれかにしてください。
- 帳票様式定義体の定義体名は、最大64バイトです。

なお、帳票の形式はすべて利用可能です。

(フリーフレーム形式、カード形式、伝票形式、集計表形式、矩形繰返し形式、タックシール形式)

## 出力可能な文字コード(ローカルに出力する場合)

- Windowsの場合

表B.30 Information Integratorで利用可能なInterstage List Creatorの文字コード(Windows)

文字コード		Interstage List Creatorで利用可能な文字コード			
		Shift-JIS	Unicode (UCS2 LE)	Unicode (UCS2 BE)	Unicode (UTF8)
Shift-JIS		○			
Unicode	UCS2			△	
	UTF8				△
	UTF8-4				
	UCS2(MS)		△		
	confused				
	confused(MS)				
	UTF16				
	UTF16(MS)				
EUC	U90				
	S90				
	HP				
JEF-EBCDIC	カナ(領域重視)				
	ASCII(領域重視)				
	カナ(字形重視)				
	ASCII(字形重視)				
IBM日本語文字 セット	カナ				
	英小文字				

○:利用可能

△:シングルフォームでの出力時のみ利用可能

- Linuxの場合

表B.31 Information Integratorで利用可能なInterstage List Creatorの文字コード(Linux)

文字コード	Interstage List Creatorで利用可能な文字コード					
	Shift-JIS	Unicode (UCS2 LE)	Unicode (UCS2 BE)	Unicode (UTF8)	EUC (U90)	EUC (S90)
Shift-JIS						
Unicode	UCS2					
	UTF8			○		
	UTF8-4					
	UCS2(MS)					
	confused					
	confused(MS)					
	UTF16					
	UTF16(MS)					
EUC	U90				△	
	S90					△
	HP					
JEF-EBCDIC	カナ (領域重視)					
	ASCII (領域重視)					
	カナ (字形重視)					
	ASCII (字形重視)					
IBM日本語文字セット	カナ					
	英小文字					

○:利用可能

△:シングルフォームでの出力時のみ利用可能

- Solarisの場合

表B.32 Information Integratorで利用可能なInterstage List Creatorの文字コード(Solaris)

文字コード	Interstage List Creatorで利用可能な文字コード					
	Shift-JIS	Unicode (UCS2 LE)	Unicode (UCS2 BE)	Unicode (UTF8)	EUC (U90)	EUC (S90)
Shift-JIS	△					
Unicode	UCS2					
	UTF8			△		
	UTF8-4					
	UCS2(MS)					
	confused					
	confused(MS)					
	UTF16					



文字コード		Interstage List Creatorで利用可能な文字コード					
		Shift-JIS	Unicode(UCS2 LE)	Unicode(UCS2 BE)	Unicode(UTF8)	EUC(U90)	EUC(S90)
	UTF16(MS)						
EUC	U90					△	
	S90						○
	HP						
JEF-EBCDIC	カナ(領域重視)						
	ASCII(領域重視)						
	カナ(字形重視)						
	ASCII(字形重視)						
IBM日本語文字セット	カナ						
	英小文字						

○:利用可能

△:シングルフォームでの出力時のみ利用可能

**出力可能な文字コード(リモートに出力する場合)**

Information Integratorが動作するシステム	帳票出力先システム	利用可能な文字コード	
Windows	Windows	Shift-JIS	
		UNICODE(UCS2 LE)	
		UNICODE(UCS2 BE)	
		UNICODE(UTF-8)	
	Linux	UNICODE(UTF-8)	
	Solaris		Shift-JIS
UNICODE(UTF-8)			
Linux	Windows	Shift-JIS	
		UNICODE(UCS2 LE)	
		UNICODE(UCS2 BE)	
		UNICODE(UTF-8)	
	Linux		UNICODE(UTF-8)
			EUC(U90)
			EUC(S90)
	Solaris		UNICODE(UTF-8)
			EUC(U90)
			EUC(S90)
Solaris	Windows	Shift-JIS	
		UNICODE(UCS2 LE)	

Information Integratorが動作するシステム	帳票出力先システム	利用可能な文字コード
		UNICODE(UCS2 BE)
		UNICODE(UTF-8)
	Linux	UNICODE(UTF-8)
		EUC(U90)
		EUC(S90)
	Solaris	Shift-JIS
		UNICODE(UTF-8)
		EUC(U90)
		EUC(S90)

### 出力可能なデータ型

可能なデータ型の対応を以下の表に示します。

表B.33 出力可能なデータ型、およびInformation Integratorのデータ型との対応

Information Integratorのデータ型	Interstage List Creatorに出力可能な範囲	対応するInterstage List Creatorのデータ型と範囲	備考
SQL_CHAR(n) n:1~32000	SQL_CHAR(n) n:1~9999	テキスト(m) m:1~2997 矩形テキスト(m) m:1~9999 OCR-B(m) m:1~999 ラジオボタン(m) m:1~255 チェックボックス(m) m:1~255 バーコード(m) m:1~9999 組み込みメディア(m) m:1~240 郵便番号(m) m:7	n <= m の場合に配付可能。
SQL_VARCHAR(n) n:1~32000	SQL_VARCHAR(n) n:1~9999	テキスト(m) m:1~2997 矩形テキスト(m) m:1~9999 OCR-B(m) m:1~999 ラジオボタン(m) m:1~255	n <= m の場合に配付可能。

Information Integratorのデータ型	Interstage List Creatorに出力可能な範囲	対応するInterstage List Creatorのデータ型と範囲	備考
		チェックボックス(m) m:1~255 バーコード(m) m:1~9999 組み込みメディア(m) m:1~240 郵便番号(m) m:7	
SQL_NCHAR(n) n:1~16000	SQL_NCHAR(n) n:1~4999	テキスト(m) m:1~2997 矩形テキスト(m) m:1~9999 ラジオボタン(m) m:1~255 チェックボックス(m) m:1~255	(nの各文字コードでのバイト数換算)<=mの場合に配付可能。
SQL_VARCHAR(n) n:1~16000	SQL_VARCHAR(n) n:1~4999	テキスト(m) m:1~2997 矩形テキスト(m) m:1~9999 ラジオボタン(m) m:1~255 チェックボックス(m) m:1~255	(nの各文字コードでのバイト数換算)<=mの場合に配付可能。
SQL_INTEGER	SQL_INTEGER	数字(p2,q2) p2:10~18 q2:0	
SQL_SMALLINT	SQL_SMALLINT	数字(p2,q2) p2:10~18 q2:0	
SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	SQL_DECIMAL(p1,q1) p1:1~18 q1:0~p1	数字(p2,q2) p2:1~18 q2:0~p2	p1<=p2、q1<=q2の場合に配付可能。
SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	SQL_NUMERIC(p1,q1) p1:1~18 q1:0~p1	数字(p2,q2) p2:1~18 q2:0~p2	p1<=p2、q1<=q2の場合に配付可能。
SQL_REAL	×		

Information Integratorのデータ型	Interstage List Creatorに出力可能な範囲	対応するInterstage List Creatorのデータ型と範囲	備考
SQL_FLOAT	×		
SQL_DATE	SQL_DATE	テキスト(m) m:10～2997 矩形テキスト(m) m:10～9999	
SQL_TIME	SQL_TIME	テキスト(m) m:8～2997 矩形テキスト(m) m:8～9999	
SQL_TIMESTAMP	SQL_TIMESTAMP	テキスト(m) m:19～2997 矩形テキスト(m) m:19～9999	
SQL_INTERVALTIME	×		
SQL_LONGVARIABLE(n) n:1～2147483647	×		

×:利用不可

## B.2.7 Information Integratorで配付できるInformation Storage形式

Information Integratorで配付できるInformation Storage形式は、収集できるInformation Storage形式と同じです。

利用可能な形式については、“[B.1.6 Information Integratorで収集できるInformation Storage形式](#)”を参照してください。

### 配付時のファイル名

配付するデータは、Information Storageの設定(カテゴリ定義)に応じて、それぞれ以下のファイル名でInformation Storageに格納されます。

カテゴリ定義(ファイル名構成要素)	格納されるファイル名*
入力データファイル名	プレフィックス_III一意データファイル名
起動時指定ファイル名	プレフィックス_IIS起動時指定ファイル名
登録日時+枝版	プレフィックス_登録日時_枝版
枝版+登録日時	プレフィックス_枝版_登録日時
入力データファイル名+登録日時	プレフィックス_III一意のファイル名_登録日時
起動時指定ファイル名+登録日時	プレフィックス_IIS起動時指定ファイル名_登録日時
入力データファイル名+枝版	プレフィックス_III一意のファイル名_枝版
起動時指定ファイル名+枝版	プレフィックス_IIS起動時指定ファイル名_枝版
入力データファイル名+枝版+登録日時	プレフィックス_III一意のファイル名_枝版_登録日時
起動時指定ファイル名+枝版+登録日時	プレフィックス_IIS起動時指定ファイル名_枝版_登録日時
入力データファイル名+登録日時+枝版	プレフィックス_III一意のファイル名_登録日時_枝版
起動時指定ファイル名+登録日時+枝版	プレフィックス_IIS起動時指定ファイル名_登録日時_枝版

\*:各ファイル名に含まれる「\_III」、「\_IIS」の文字列以外の情報は、カテゴリ定義(ファイル名構成要素)で指定する情報です。  
 カテゴリ定義の詳細については、“Information Storage”のマニュアルを参照してください。

## B.2.8 Information Integratorで配付できるBLOB形式

Information Integratorでは、すべてのBLOBデータをAzure Storageへ配付することができます。

## B.3 Information Integratorで対応するデータ型のデータ例

Information Integratorで対応するデータ型のデータ例は、以下のとおりです。

No	Information Integratorのデータ型とデータ例	
	データ型	データ例 [上段:テキスト表記] [下段:16進表記]
1	SQL_CHAR: 英数字/日本語文字	1234 (0x31323334)
2	SQL_CHAR: 英数字/日本語文字 数値文字列(先頭空白詰め、または先頭0詰め) ※1	□□1234 (0x202031323334) 001234 (0x303031323334)
3	SQL_CHAR: 英数字/日本語文字 数値文字列カンマ付(先頭空白詰め、または先頭0詰め) ※2	□□1,234 (0x20312c323334) 01.234 (0x30312c323334)
4	SQL_VARCHAR: 可変長英数字/日本語文字	1234 (0x040031323334)
5	SQL_NCHAR: 日本語文字	1234 (0x8250825182528253)
6	SQL_VARNCHAR: 可変長日本語文字	1234 (0x04008250825182528253)
7	SQL_INTEGER: 整数 10桁(4バイト領域)	《表示できません》 (0xD2040000)
8	SQL_SMALLINT: 整数 5桁(2バイト領域)	《表示できません》 (0xD204)
9	SQL_NUMERIC: 整数、小数点付き外部10進数 ・ 符号なし ・ 符号あり、末尾、ビット形式 ・ 符号あり、先頭、ビット形式 ・ 符号あり、先頭、文字形式 ・ 符号あり、末尾、文字形式	1234 (0x31323334) 123D (0x31323344) A234 (0x41323334) +1234 (0x2B31323334) 1234+ (0x313233342B)

No	Information Integratorのデータ型とデータ例	
	データ型	データ例 [上段:テキスト表記] [下段:16進表記]
10	SQL_DECIMAL: 整数、小数点付き内部10進数 ・符号あり	《表示できません》 (0x01234c)
11	SQL_REAL: 整数、小数点付き数字、指数 4バイト領域の単精度浮動小数点	《表示できません》 (0x449a4000)
12	SQL_FLOAT: 整数、小数点付き数字、指数 8バイト領域の倍精度浮動小数点	《表示できません》 (0x4093480000000000)
13	SQL_FLOAT: 整数、小数点付き数字、指数 8バイト領域の倍精度浮動小数点	《表示できません》 (0x4093480000000000)
14	SQL_FLOAT: 整数、小数点付き数字、指数 8バイト領域の倍精度浮動小数点	《表示できません》 (0x4093480000000000)
15	SQL_DATE: 日付	2010@0408 (0x3230314030343038)
16	SQL_TIME: 時間	221513 (0x323231353133)
17	SQL_TIMESTAMP: 日付時間	201@0408221513 (0x3230314030343038323231353133)
18	SQL_LONGVARIABLE: バイナリオブジェクト(長さ部あり)	《表示できません》 (0x00000000020000001234)
19	SQL_BINARY: バイナリオブジェクト(長さ部なし)	《表示できません》 (0x0102030405)
20	SQL_BIT: ビット	《表示できません》 全体桁を '8' と定義している場合:  00001010 (0x0a)

※1:データ変換定義に以下の2つの指定をすることで変換可能です。

「演算式-演算種別」に「数字表記変換」

「演算式-演算種別」に「パディング」、「演算式-引数」に「左,空白」または「左,0」

※2:データ変換定義の「演算式-演算種別」に「数字表記変換」を指定することで、データを数字として扱うことが可能です。ただし、カンマ付き数字を出力することはできません。

## B.4 変換処理できる項目属性の一覧

表B.34 変換処理できる項目属性の一覧

項目属性(入力側)		フォーマット変換	文字コード変換	項目並び替え/項目選択	項目追加	属性変換	トリム	パディング	文字列の部分抽出	文字列の設定/文字列の連結	大文字/小文字変換	全角/半角変換	ひらがな/カタカナ変換	濁点・半濁点変換	四則演算	項目条件判定	数字表記変換	日時の書式変換	日時の部分取得	振分け(キー)	ソート(キー)	連結(キー)	集計(キー)
文字属性(英数字属性)	固定長形式	SQL_CHARACTER	○	○	○	○	○	○	○	○	○注1	○注2	○注4	○	○注5	○	○	×	×	○	○	○	×
	可変長形式	SQL_VARCHAR	○	○	○	○	○	○	○	○	○注1	○注2	○注4	○	○注5	○	○	×	×	○	○	×	×
文字属性(日本語属性)	固定長形式	SQL_NCHAR	○	○	○	○	×	○	×	○	×	○注2、3	○注4	×	○注5	○	×	×	×	○	○	○	×
	可変長形式	SQL_VARCHAR	○	○	○	○	×	○	×	○	×	○注2、3	○注4	×	○注5	○	×	×	×	○	○	×	×
数値属性	整数属性	SQL_INTEGER	○	○	○	○	×	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○
		SQL_SMALLINT	○	○	○	○	○	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○
	内部10進数	SQL_DECIMAL	○	○	○	○	○	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○

項目属性(入力側)	フォーマット変換	文字コード変換	項目並び替え/項目選択	項目追加	属性変換	トリム	パディング	文字列の部分抽出	文字列の設定/文字列の連結	大文字/小文字変換	全角/半角変換	ひらがな/カタカナ変換	濁点・半濁点変換	四則演算	項目条件判定	数字表記変換	日時の書式変換	日時の部分取得	振分け(キー)	ソート(キー)	連結(キー)	集計(キー)
外部10進数属性	SQL_NUMERIC	○	○	○	○	○	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○
浮動小数点属性(単精度浮動小数点属性)	SQL_REAL	○	○	○	○	○	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○
浮動小数点属性(倍精度浮動小数点属性)	SQL_FLOAT	○	○	○	○	○	×	×	×	×	×	×	×	×	○	×	×	×	○	○	○	○



項目属性(入力側)		フォーマット変換	文字コード変換	項目並び替え/項目選択	項目追加	属性変換	トリム	パディング	文字列の部分抽出	文字列の設定/文字列の連結	大文字/小文字変換	全角/半角変換	ひらがな/カタカナ変換	濁点・半濁点変換	四則演算	項目条件判定	数字表記変換	日時の書式変換	日時の部分取得	振分け(キー)	ソート(キー)	連結(キー)	集計(キー)
日付属性	SQL_DATE	○	○	○	○	○	×	×	×	×	×	×	×	×	×	○	×	○	○	○	×	×	×
日付時間属性	SQL_TIME	○	○	○	○	○	×	×	×	×	×	×	×	×	×	○	×	○	○	○	×	×	×
時間属性	SQL_TIMESTAMP	○	○	○	○	○	×	×	×	×	×	×	×	×	×	○	×	○	○	○	×	×	×
時間間隔属性	SQL_INTERVALTIME	×	×	○	×	○	×	×	×	×	×	×	×	×	×	○	×	×	×	×	×	×	×
バイナリオブジェクト属性	長さ部あり SQL_LONGVARCHAR	○注6	×	○	○	×	×	×	×	×	×	×	×	×	×	○	×	×	×	×	×	×	×
	長さ部 SQL_BINARY	○注7	×	○	○	×	×	×	×	×	×	×	×	×	×	○	×	×	×	×	×	×	×

項目属性(入力側)	フォーマット変換	文字コード変換	項目並び替え/項目選択	項目追加	属性変換	トリム	パディング	文字列の部分抽出	文字列の設定/文字列の連結	大文字/小文字変換	全角/半角変換	ひらがな/カタカナ変換	濁点・半濁点変換	四則演算	項目条件判定	数字表記変換	日時の書式変換	日時の部分取得	振分け(キ一)	ソート(キ一)	連結(キ一)	集計(キ一)
ビット属性	SQL BIT	○	×	○	○	×	×	×	×	×	×	×	×	×	○	×	×	×	×	×	×	×

注1)半角英字以外は不可(ASCIIのみサポート)

注2)ひらがな、漢字の場合は不可(半角に相当するものが存在する場合に限る)

注3)半角→全角は不可。全角→半角への変換は、半角が「NCHAR」に格納されてしまうため、Symfoware Serverでは利用できません。

注4)ひらがな、カタカナ以外の場合は不可

注5)全角ひらがな、全角カタカナ以外の場合は不可

注6)変換後のフォーマットがCSV形式の場合、BASE64形式に変換

注7)変換後のフォーマットがCSV形式の場合、BASE64形式に変換

## B.5 CSVファイルで扱えるデータ属性

CSVファイル形式でサポート可能な文字列の長さや数値の範囲を以下の表に示します。



データ構造定義で以下のデータ形式を選択した場合、サポート可能な文字列の長さや数値の範囲については、XMLファイルで扱えるデータ属性と同一の範囲となります。詳細は、「[B.7 XMLファイルで扱えるデータ属性](#)」を参照してください。

- ・ 一般(繰り返しあり)
- ・ 伝票形式

### 変換可能な値の範囲

表B.35 扱えるデータの範囲

属性	相互変換可能な値	備考
SQL_INTEGER	-2147483648～2147483647	
SQL_SMALLINT	-32768～32767	
SQL_NUMERIC (p,q)	有効桁数18桁	zone形式
SQL_DECIMAL (p,q)	有効桁数18桁	pack形式

属性	相互変換可能な値	備考
SQL_REAL	IEEE ± 1.17E-38~3.40E+38  有効桁数約8桁	
SQL_FLOAT	IEEE ± 2.22E-308~1.79E+308  有効桁数約17桁	
SQL_CHAR	32000文字	
SQL_VARCHAR	31998文字	
SQL_NCHAR	16000文字	Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。 Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_VARNCHAR	15999文字	Unicode(utf8)では3バイトコードが中心のため、最大長10666文字として計算してください。 Unicode(utf8-4)では最大4バイトコードとなるため、最大長8000文字として計算してください。
SQL_DATE	YYYY/MM/DD	
SQL_TIME	HH:MM:SS	
SQL_TIMESTAMP	YYYY/MM/DD HH:MM:SS	
SQL_LONGVARBINARY	2147483647バイト	base64形式のみ 「JEF-EBCDIC カナ(領域重視または字形重視)」以外の文字コードでのみ利用可能です。
SQL_BINARY	2147483647バイト	base64形式のみ 「JEF-EBCDIC カナ(領域重視または字形重視)」以外の文字コードでのみ利用可能です。
SQL_BIT	128ビット(列桁数)	バイナリデータでは、指定された桁数に対して、以下のサイズとなります。 1桁~8桁まで:1バイト 9桁~16桁まで:2バイト ... 121桁~128桁まで:16バイト

## 注意

- 変換後のデータをデータベースに格納する場合は、各データベースにおいて格納可能な範囲を確認してください。
- 表計算ソフトウェアおよびデータベースで使用可能な範囲は、各製品の説明書を参照してください。

## CSV形式データにおいて扱える数値表現

CSVファイル形式からバイナリファイル形式へ変換する場合に利用可能な数値表現を以下の表に示します。

表B.36 CSV形式データにおいて扱える数値表現

属性	扱える数値表現	備考
SQL_INTEGER	整数	10桁
SQL_SMALLINT	整数	5桁
SQL_NUMERIC	整数、小数点付き数字	

属性	扱える数値表現	備考
SQL_DECIMAL	整数、小数点付き数字	
SQL_REAL	整数、小数点付き数字、指数	
SQL_FLOAT	整数、小数点付き数字、指数	

### CSVファイル形式で扱える数値の形式

- ・ 小数点以上の桁は、定義された桁数と必ず一致する必要があります。
- ・ 小数点以下の桁が定義された桁数より大きい場合、桁あふれとして扱います。
- ・ 符号は正の場合「+」、負の場合「-」で、設定が可能です。符号を省略した場合は、正として扱います。

【例】zone(5.2)の場合

-012.10	正常
012.1	正常
012	正常
012.100	桁あふれ
12.	設定値異常

- ・ 桁の区切り記号は指定できません。

【例】intの場合

123456789	正常
123,456,789	設定値異常

### CSVファイル形式で扱える日付、時間および日付時間表現

属性	扱える表現	備考
SQL_DATE	YYYY/[M]M/[D]Dまたは、YYYY-[M]M-[D]D	[]内は省略可
SQL_TIME	[H]H:[M]M:[S]S	[]内は省略可
SQL_TIMESTAM P	YYYY/[M]M/[D]D [H]H:[M]M:[S]Sまたは、 YYYY-[M]M-[D]D [H]H:[M]M:[S]S	[]内は省略可



#### 注意

- ・ 日付時間属性の日付と時間の間に半角の空白文字が1文字必要です。
- ・ 区切り文字が空白文字の時、日付時間属性は利用できません。

### 扱える最大の行の長さについて

扱える行の最大の長さは、以下のとおりです。

- ・ バイナリファイル形式の表現で、32000バイトです。
- ・ CSVファイル形式の表現で、64000バイトです。

### 扱える最大の列数

扱える最大の列数は、32000です。

## 長さが0の可変長文字列データの扱いについて

バイナリファイル形式からCSVファイル形式への変換で、長さが0の可変長文字列データの扱いは、以下のとおりです。

- ・ 列属性がvcharの場合、半角空白を1つ設定します。
- ・ 列属性がvncharの場合、全角空白を1つ設定します。

## B.6 バイナリファイルで扱えるデータ属性

バイナリファイル形式では、すべてのデータ属性を利用可能です。

ただし、文字コード「JEF-EBCDIC カナ(領域重視または字形重視)」の場合は、データ属性が「SQL\_LONGVARIABLE」または「SQL\_BINARY」を利用できません。

また、データ構造定義で以下のデータ形式を選択した場合、サポート可能な文字列の長さや数値の範囲については、XMLファイルで扱えるデータ属性と同一の範囲となります。詳細は、「[B.7 XMLファイルで扱えるデータ属性](#)」を参照してください。

- ・ 一般(繰り返しあり)
- ・ 伝票形式

## B.7 XMLファイルで扱えるデータ属性

XMLファイル形式でサポート可能な文字列の長さや数値の範囲を以下の表に示します。

表B.37 扱えるデータの範囲

属性	相互変換可能な値
SQL_INTEGER	-2147483648～2147483647
SQL_SMALLINT	-32768～32767
SQL_NUMERIC(p,q)	有効桁数18桁
SQL_DECIMAL(p,q)	有効桁数18桁
SQL_CHAR	32000バイト

## B.8 列属性について

以下の各変換で、変換可能な列属性について説明します。

- ・ CSVファイル形式からバイナリファイル形式への変換
- ・ バイナリファイル形式からCSVファイル形式への変換
- ・ XMLファイル形式からバイナリファイル形式への変換
- ・ バイナリファイル形式からXMLファイル形式への変換
- ・ XMLファイル形式からCSVファイル形式への変換
- ・ CSVファイル形式からXMLファイル形式への変換

それぞれ以下の表に示します。

表B.38 CSVファイル形式からバイナリファイル形式へ変換可能な列属性

CSVファイル形式の値	テンプレートに指定した属性情報	変換後のバイナリファイル形式の属性	備考
英数字文字	SQL_CHAR	SQL_CHAR	
	SQL_VARCHAR	SQL_VARCHAR	
日本語文字	SQL_CHAR	SQL_CHAR	
	SQL_VARCHAR	SQL_VARCHAR	

CSVファイル形式の値	テンプレートに指定した属性情報	変換後のバイナリファイル形式の属性	備考
	SQL_NCHAR	SQL_NCHAR	
	SQL_VARNCHAR	SQL_VARNCHAR	
整数	SQL_INTEGER	SQL_INTEGER	
	SQL_SMALLINT	SQL_SMALLINT	
小数点付き数字	SQL_NUMERIC	SQL_NUMERIC	符号は正:4、負:5
	SQL_DECIMAL	SQL_DECIMAL	符号は正:C、負:D
	SQL_REAL	SQL_REAL	
	SQL_FLOAT	SQL_FLOAT	
日付	SQL_DATE	SQL_DATE	
	SQL_CHAR	SQL_CHAR	
時間	SQL_TIME	SQL_TIME	
	SQL_CHAR	SQL_CHAR	
日付時間	SQL_TIMESTAMP	SQL_TIMESTAMP	
	SQL_CHAR	SQL_CHAR	

表B.39 バイナリファイル形式からCSVファイル形式へ変換可能な列属性の対応

バイナリファイル形式の形式属性	テンプレートに指定した属性情報	変換後のCSVファイル形式の値
SQL_CHAR	SQL_CHAR	英数字文字
SQL_VARCHAR	SQL_VARCHAR	
SQL_NCHAR	SQL_NCHAR	日本語文字
SQL_VARNCHAR	SQL_VARNCHAR	
SQL_INTEGER	SQL_INTEGER	整数
SQL_SMALLINT	SQL_SMALLINT	
SQL_NUMERIC	SQL_NUMERIC	小数点付き数字
SQL_DECIMAL	SQL_DECIMAL	
SQL_REAL	SQL_REAL	
SQL_FLOAT	SQL_FLOAT	
SQL_DATE	SQL_DATE	日付
SQL_TIME	SQL_TIME	時間
SQL_TIMESTAMP	SQL_TIMESTAMP	日付時間

表B.40 XMLファイル形式からバイナリファイル形式へ変換可能な列属性

XMLファイル形式の値	テンプレートに指定した属性情報	変換後のバイナリファイル形式の属性	備考
英数字文字	SQL_CHAR	SQL_CHAR	
日本語文字	SQL_CHAR	SQL_CHAR	
整数	SQL_INTEGER	SQL_INTEGER	
	SQL_SMALLINT	SQL_SMALLINT	
小数点付き数字	SQL_NUMERIC	SQL_NUMERIC	符号は正:4、負:5

XMLファイル形式の値	テンプレートに指定した属性情報	変換後のバイナリファイル形式の属性	備考
	SQL_DECIMAL	SQL_DECIMAL	符号は正:C、負:D

表B.41 バイナリファイル形式からXMLファイル形式へ変換可能な列属性の対応

バイナリファイル形式の形式属性	テンプレートに指定した属性情報	変換後のXMLファイル形式の値
SQL_CHAR	SQL_CHAR	英数字文字
SQL_INTEGER	SQL_INTEGER	整数
SQL_SMALLINT	SQL_SMALLINT	
SQL_NUMERIC	SQL_NUMERIC	小数点付き数字
SQL_DECIMAL	SQL_DECIMAL	

表B.42 XMLファイル形式からCSVファイル形式へ変換可能な列属性

XMLファイル形式の値	テンプレートに指定した属性情報	変換後のCSVファイル形式の属性	備考
英数字文字	SQL_CHAR	SQL_CHAR	
日本語文字	SQL_CHAR	SQL_CHAR	
整数	SQL_INTEGER	SQL_INTEGER	
	SQL_SMALLINT	SQL_SMALLINT	
小数点付き数字	SQL_NUMERIC	SQL_NUMERIC	
	SQL_DECIMAL	SQL_DECIMAL	

表B.43 CSVファイル形式からXMLファイル形式へ変換可能な列属性の対応

CSVファイル形式の形式属性	テンプレートに指定した属性情報	変換後のXMLファイル形式の値
SQL_CHAR	SQL_CHAR	英数字文字
SQL_INTEGER	SQL_INTEGER	整数
SQL_SMALLINT	SQL_SMALLINT	
SQL_NUMERIC	SQL_NUMERIC	小数点付き数字
SQL_DECIMAL	SQL_DECIMAL	

## 付録C ファイル転送エージェントで利用可能な機能一覧

ファイル転送エージェントで利用可能な機能について説明します。

Information Integratorサーバで利用できる機能を基に、ファイル転送エージェントで利用可能な機能は、以下のとおりです。

### データ収集・データ配付

機能分類	機能小分類	利用可否 ※1	備考
ファイル受信・ファイル送信	—	△	Information Integratorサーバとの接続、かつ、接続プロトコルがFTP+のみ利用可。
ファイル入力・ファイル出力	—	○	
データベース形式	Symfoware Server	○	
	Oracle	○	
	SQL Server	○	
	DB2	○	
	PostgreSQL	○	
	SQL Azure	×	
	その他のデータベース(ODBC接続)	○	
オブジェクト形式	Salesforceオブジェクト	×	
Information Storage形式	Information Storage	×	
BLOB形式	Azure Storage	×	
サービスバス連携形式 (Service Integrator形式)	Service Integrator	×	
帳票出力	—	×	
Excelデータ形式	—	○	

※1) ○:利用可能、△:条件付きで利用可能、×:利用不可

### その他の機能

機能分類	機能小分類	利用可否 ※1	備考
定義作成/管理	定義作成	○	統合管理オプションとファイル転送エージェントを併用する場合、処理プロセスの設計情報は、Information Integratorサーバで定義の作成および管理を行います。ファイル転送エージェントで利用する定義は、Information Integratorサーバで作成し、ファイル転送エージェントに配付して利用します。
	定義管理	○	
イベント監視	イベント監視	△	監視対象のディレクトリにファイルが生成されたときに後続の処理が実行される形態のみ利用可
相手側ジョブ起動	—	×	

※1) ○:利用可能、△:条件付きで利用可能、×:利用不可



上記機能以外は、Information Integratorサーバで利用可能な機能と同等な機能が利用できます。

## 付録D フォーマット形式

Information Integratorで対応するファイル形式のフォーマットについて説明します。

- 一般フォーマット  
1レコードの中に複数の項目がある固定長のフォーマットです。
- 一般フォーマット(繰返しあり)  
一般フォーマットで、レコードの中に「集団項目」をもち、集団項目に「繰返し」を定義するフォーマットです。
- レコード終端可変長形式フォーマット  
レコード終端可変長形式フォーマットとは、「ヘッダ部」と「明細部」で構成されており、レコード終端まで明細部が繰り返されるフォーマットです。
- 繰返し数指定可変長フォーマット  
繰返し数指定可変長フォーマットとは、「ヘッダ部」と「明細部」で構成されており、明細部が繰り返されるフォーマットです。
- 伝票形式フォーマット  
伝票形式フォーマットとは、「ヘッダレコード」、「明細レコード」、「トレーラレコード」という異なるレイアウトの順に構成される、固定長のレコードの集合体です。

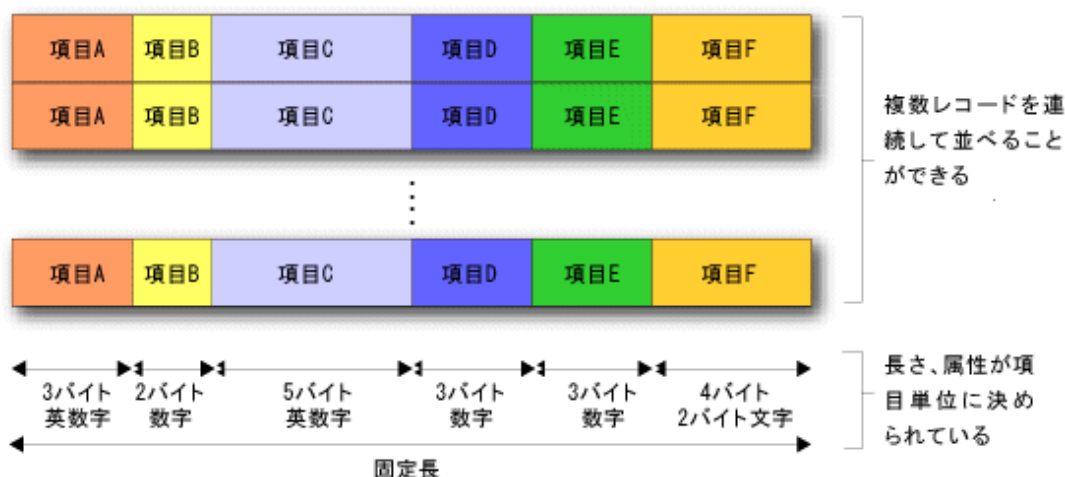
### ポイント

- 1レコードは、フォーマットの単位データを指します。レコードの項目をフィールドと呼びます。複数レコードを一度に入出力することも可能です。
- 一般フォーマット(集団項目の繰返しあり)、レコード終端可変長形式フォーマット、繰返し数指定可変長フォーマットは、いずれも定義の「一般(繰返しあり)」に指定して利用するフォーマット形式です。

## D.1 一般フォーマット

一般フォーマットとは、1レコードの中に複数の項目がある固定長のフォーマットのことで、各項目に、長さ、属性が決められていて、項目の順番も入れ替わることなく並んでいる必要があります。

図D.1 一般フォーマット

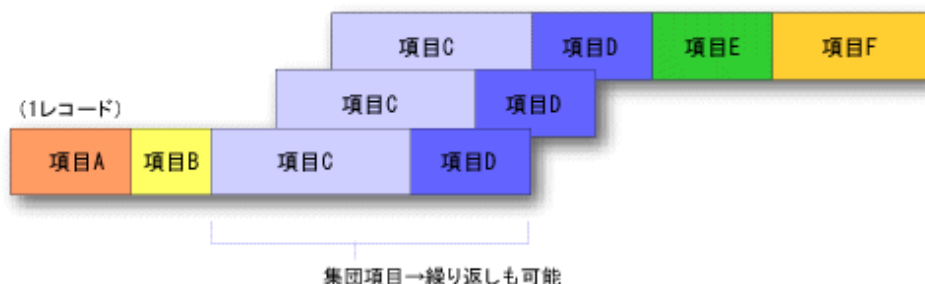


定義方法の詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。

## D.2 一般フォーマット(繰返しあり)

一般フォーマットで、レコードの中に「集団項目」をもつことができます。集団項目には「繰返し」を定義することができます。

図D.2 集団項目について



上記の場合、項目Cと項目Dが3回順に繰返しています。項目Cと項目Dをグループ化しており、これを1つの集団項目と呼びます。

### P ポイント

- ・ 1レコード内に定義できる集団項目数の最大は1,024個です。
- ・ 集団項目の繰返し数の最大は1,024回です。
- ・ 集団項目の繰返し階層(ネストレベル)の最大は99階層です。

### 定義概要

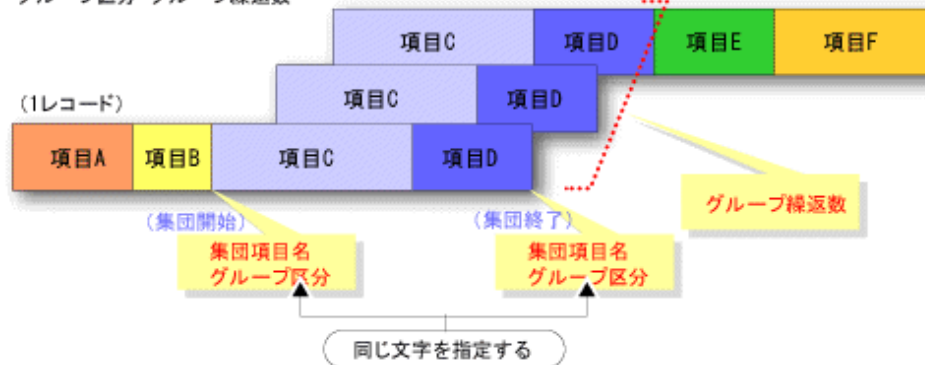
以下の内容をデータ構造定義に定義します。

- ・ 繰返しグループ名
- ・ 繰返し階層
- ・ 繰返し回数

集団項目として扱う項目すべてに「繰返しグループ名」をデータ構造定義に指定します。「繰返しグループ名」を指定した項目は、繰返し階層を指定し、どの階層レベルかを宣言してください。また、同じ繰返しグループの先頭の項目に集団項目の「繰返し回数」を指定します。

図D.3 集団項目の定義

グループ区分・グループ繰返数



図D.4 集団項目定義の例

集団項目ヘッダ

繰り返し設定

項目名	項目A	項目B	項目C	...	項目B	項目C	項目D
属性	SQL_INT	SQL_CHAR	SQL_CHAR	...	SQL_CHAR	SQL_CHAR	SQL_CHAR
桁数	4	8	10	...	8	10	6

定義方法の詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。

## D.3 レコード終端可変長形式フォーマット

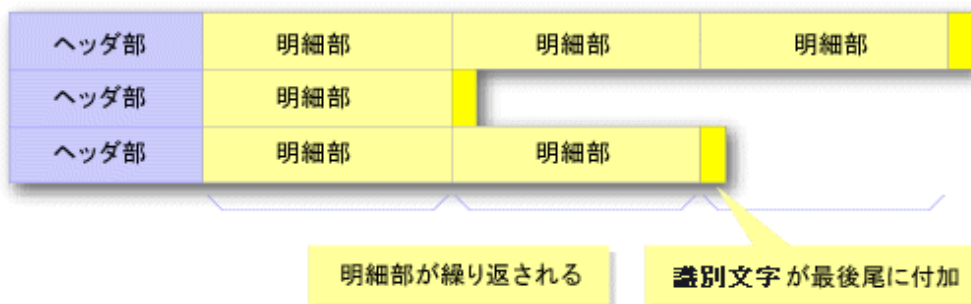
レコード終端可変長フォーマットとは、「ヘッダ部」、「明細部」、「レコード終端文字」で構成されており、明細部が繰り返されるフォーマットです。

明細部の繰り返しは無制限で、レコードごとに繰り返し数が異なってもかまいません。

ただし、繰り返しの最終明細部の直後に、ここで明細部が終わることを示す識別記号(レコード終端文字)が付加されている必要があります。識別記号には、「改行マーク」(UNIXの場合は、「LF」、Windowsの場合は「CR+LF」)などを使用します。

識別記号と同じ文字列が、データとして明細部のデータ中に現れても問題ありませんが、明細部の先頭には現れてはなりません。なお、レコード長に識別記号は含まれません。

図D.5 レコード終端可変長フォーマット



### 定義概要

以下の内容を定義します。

- ・ 繰り返し開始項目(データ構造定義)
- ・ レコード終端文字(フォーマット定義)

どの項目から繰り返しが始まるかを「繰り返し開始項目」に指定します。「繰り返し開始項目」からデータ構造定義に定義した最後の項目までの範囲が上述の明細部です。レコード終端の判別は、CSV形式のデータの場合、改行コード(Windows: CRLF、Solaris/Linux: LF)までとなります。バイナリ形式のデータの場合、フォーマット定義のレコード終端文字の指定値で終端を判別します。

定義方法の詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。

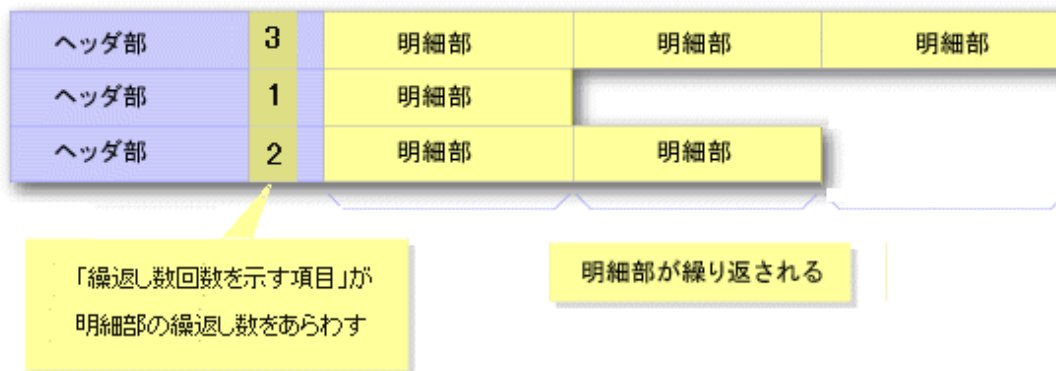
## D.4 繰り返し数指定可変長フォーマット

繰り返し数指定可変長フォーマットとは、「ヘッダ部」と「明細部」で構成されており、明細部が繰り返されるフォーマットです。

明細部の繰り返しは無制限で、レコードごとに繰り返し数が異なってもかまいません。

ここまでは「レコード終端可変長フォーマット」と同じですが、改行マークは不要で、代わりに「ヘッダ部」の1つの項目に「明細部の繰り返し回数を示す項目」をもち、その値が「明細部」の繰り返し数を表します。

図D.6 繰返し数指定可変長フォーマット



**注意**

明細部の中にさらに明細部をもつことはできません。また異なる明細部を1レコードの中に複数箇所もつこともできません。

**定義概要**

以下の内容を定義します。

- ・ 繰返し数設定項目(データ構造定義)
- ・ 繰返し開始項目(データ構造定義)

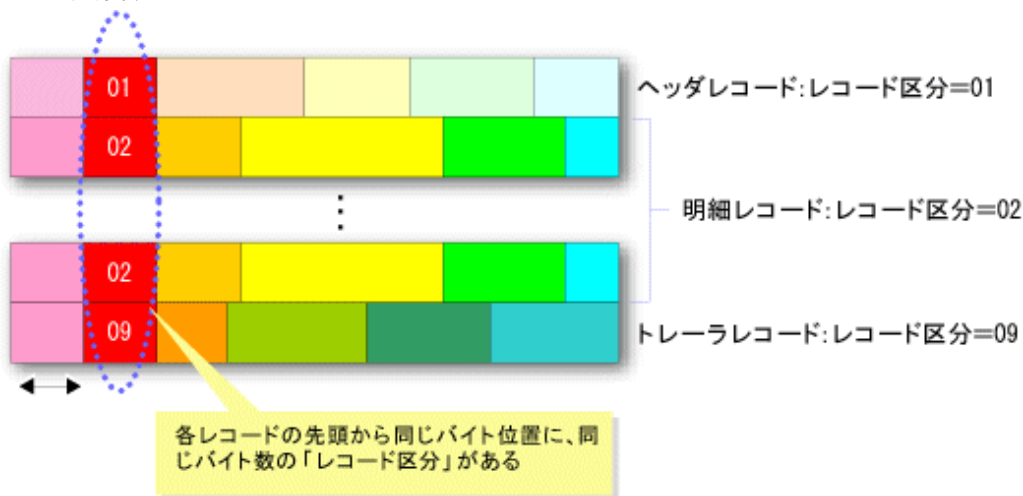
繰返し数が設定されている項目を「繰返し数設定項目」に指定します。また、どの項目から繰返しが始まるかを「繰返し開始項目」に指定します。「繰返し開始項目」からデータ構造定義に定義した最後の項目までの範囲が上述の明細部です。

定義方法の詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。

**D.5 伝票形式フォーマット**

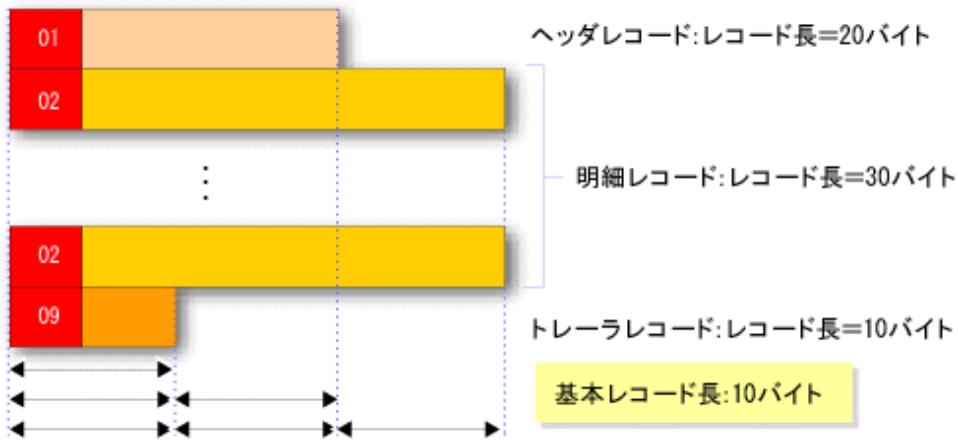
伝票形式フォーマットとは、「ヘッダレコード」、「明細レコード」、「トレーラレコード」という異なるレイアウトの順に構成される、固定長のレコードの集合体です。これらのレコードのひとつまとまりを1伝票と呼びます。

図D.7 伝票識別子について

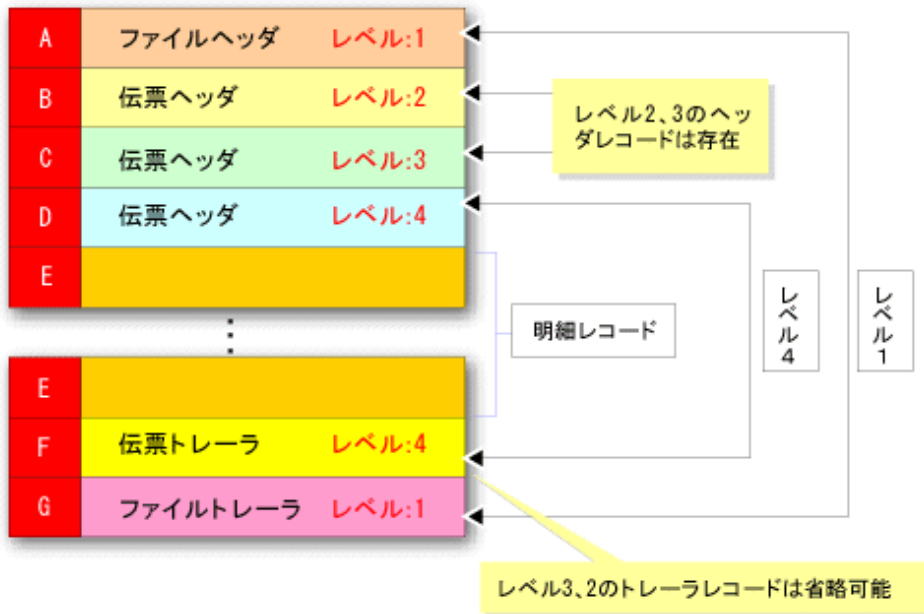


ヘッダレコード、明細レコード、トレーラレコードを識別するための値を「伝票識別子」といいます。ヘッダレコード、明細レコード、トレーラレコードには、それぞれの先頭から同じバイト位置に同じバイト数の伝票識別子があり、その値の違いによりレコードの違いを判別します。

図D.8 基本レコード長について



図D.9 レベルについて



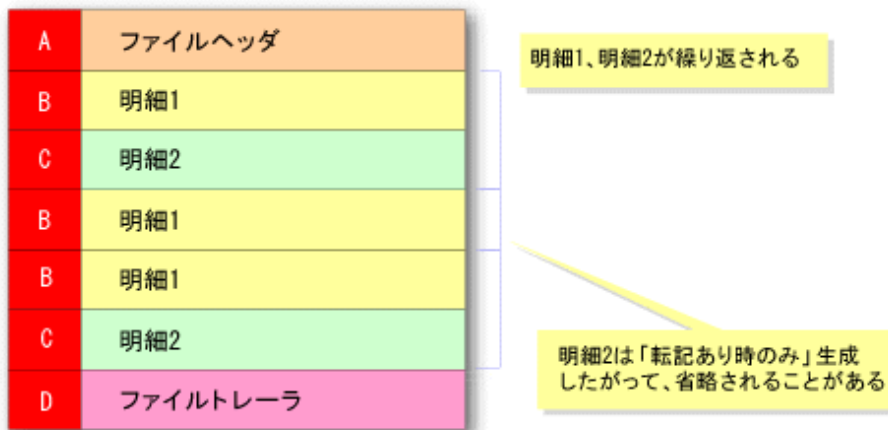
各レコードの階層構造を示す値を「レベル」といいます。

ヘッダレコード、明細レコード、トレーラレコードは複数の種類を定義することができます。また、ヘッダレコードとトレーラレコードはペアで定義します。なお、トレーラレコードは省略可能なので、ヘッダレコードとトレーラレコードの数は一致していません。

先頭の階層のヘッダレコードを「ファイルヘッダ」、2番目以降を「伝票ヘッダ」と呼びます。ヘッダレコードにはそれぞれレベルをつけて管理します。レベルには先頭階層から順に、1～9を設定します。したがって、ファイルヘッダはレベル1で固定し、伝票ヘッダにはレベル2～9を設定してください。

同様に、トレーラレコードも、最後尾を「ファイルトレーラ」、最後尾から2番目以前を「伝票トレーラ」と呼びます。レベルはヘッダレコードと対応するので、「ファイルトレーラ」はレベル1で固定し、「伝票トレーラ」は後方から順に2～9を設定してください。

図D.10 明細レコード



明細レコードは一部を省略することができます。  
 明細レコードを複数種類定義した場合、明細レコードにはレベルがないので、複数の明細レコードは連続しています。連続している明細レコードの一部を省略可能にすることもできます。

図D.11 ブロック長



伝票形式フォーマットでは「ブロック長」を指定する必要があります。すべてのレコードの合計バイト数がブロック長の等数倍にならない場合、その差のバイト数に空白が付加されます。  
 また、伝票形式フォーマットが入力の場合は、空白を外してから、処理します。

### P ポイント

- ・ 伝票形式フォーマットの1つのレコードは、一般フォーマットと同等です。
- ・ 1つの伝票形式フォーマットの中に指定できるレコード数の最大は20個です。
- ・ 伝票識別子にはワイルドカードの“?”が指定できます。“?”は任意の1バイト文字が指定されたものと判断します。2バイト文字に使用する場合には“??”と指定します。  
 例1)“AA?”が指定された場合、“AAA”も“AAB”も対象になります。  
 例2)“小??”が指定された場合、“小一”も“小二”も対象になります。

### 定義概要

以下の内容を定義します。

- ・ 伝票種別
- ・ 伝票識別子
- ・ ブロック長

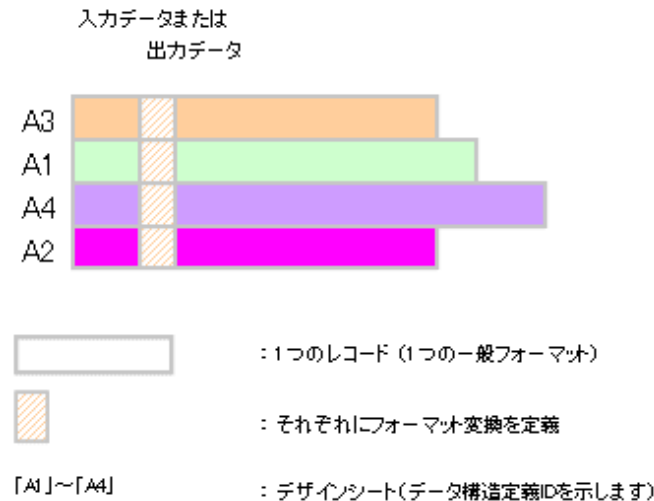
項目ごとに「伝票種別」として「ヘッダ」「明細」「トレーラ」をデータ構造定義に指定します。また、「伝票識別子」に該当する項目に識別するための値を指定します。

伝票形式の登録方法の詳細は、“III Studio ヘルプ”または、“デザインシート操作ヘルプ”を参照してください。

## D.6 複数フォーマットが混在するフォーマット

1つの入力ファイルに複数のフォーマットが混在するデータは、入力データを異なる一般フォーマットが複数連なっている形式として扱うことができます。

図D.12 混在フォーマット



### 定義概要

以下の内容を定義します。

- データ構造定義
- 変換グループID(プロセス定義)
- 出力対象とするしない(プロセス定義)

入出力データに含まれる複数のフォーマットについて、すべてのフォーマットをデータ構造定義で定義します。これらのフォーマットはデータ構造定義の[データ形式変更]ボタンで「混在フォーマット」を選択してください。

また、入出力データにどのフォーマットが含まれるかを宣言するのは、プロセス定義の「混在フォーマット変換グループID」です。1つのプロセス定義内で同一の混在フォーマット変換グループIDを指定したファンクションが、混在フォーマット対象です。なお、混在フォーマット変換グループIDは、データ変換定義のファンクションを指定します。

また、混在フォーマット出力対象は、入力データに含まれるフォーマットのデータを出力データとして出力するか否かを指定します。

### 注意

- 1つの混在フォーマットに含まれる複数のフォーマットは、形式を統一する必要があります。文字コード系、レコードタイプなどであり、複数のフォーマットはすべて同じフォーマット定義としてください。
- 混在フォーマットをフォーマットごとに分割して変換処理を行う場合、定義方法は以下のとおりです。

#### ー プロセス定義の定義方法

1. 混在フォーマットの変換ファンクションの先行処理にデータ振分ファンクションを定義します。
2. データ振分ファンクションのデータが、混在フォーマットの変換ファンクションの入力データになるように定義します。「振分データID」を入力します。入力する「振分データID」は、データ振分定義を参照してください。
3. 変換ファンクションのデータの出力有無を定義します。下記の例では、変換処理ファンクション「CONVERT1からCONVERT3」のうち、「CONVERT1」のデータを出力し、続けて、同じように「CONVERT2」のデータの出力および、「CONVERT3」のデータの出力を行っています。このような場合、必ず振分データIDが重複しないように設定してください。



一 データ振分定義の定義方法

プロセス定義のデータ振分ファンクションについて、データの振分条件を定義します。データ振分ファンクションの全件データが、混在フォーマットの変換ファンクションの入力データになるように定義します。

定義例) 混在フォーマットの変換処理が3つ存在する場合のデザインシートでの定義例を示します。

**プロセス定義シートの定義例**

項番	ファンクションID	処理の種類	先行処理の項番	混在フォーマット		振分データID
				変換グループID	出力対象とする	
1	INPUT	データ収集				
2	DIVIDE	データ指分	1			
3	CONVERT1	データ変換	2	GROUP1	する	OutData1
4	CONVERT2	データ変換	2	GROUP1	しない	OutData2
5	CONVERT3	データ変換	2	GROUP1	しない	OutData3
6	OUTPUT1	データ配信	3,4,5			
7	CONVERT1	データ変換	2	GROUP2	しない	OutData4
8	CONVERT2	データ変換	2	GROUP2	する	OutData5
9	CONVERT3	データ変換	2	GROUP2	しない	OutData6
10	OUTPUT2	データ配信	7,8,9			
11	CONVERT1	データ変換	2	GROUP3	しない	OutData7
12	CONVERT2	データ変換	2	GROUP3	しない	OutData8
13	CONVERT3	データ変換	2	GROUP3	する	OutData9
14	OUTPUT3	データ配信	11,12,13			

変換処理の先行処理で定義  
振分ファンクション DIVIDE

混在フォーマットの  
変換処理ファンクション  
CONVERT1~CONVERT3

変換ファンクションのデータを  
出力対象とするか否かを指定

振分定義シートの振分データID  
を入力

**振分定義シートの定義例**

項番	ファンクションID	振分データID
1	broadcast	OutData1
2	broadcast	OutData2
3	broadcast	OutData3
4	broadcast	OutData4
5	broadcast	OutData5
6	broadcast	OutData6
7	broadcast	OutData7
8	broadcast	OutData8
9	broadcast	OutData9

データ振分ファンクションの後続処理(データ変換処理)で  
全件データを対象にします。「broadcast」を指定します。

- 混在フォーマットを他のフォーマット(一般、一般(繰返しあり)、伝票など)に変更する場合は、データ構造(データ型や桁数)を変更したいフォーマットの形にあわせてください。

例えば、混在フォーマットを一般フォーマットに変更する場合、以下のように、データ型や桁数を一般フォーマットの形式にあわせてます。

このとき、データ変換定義での定義は、変換前データも、変換後データも、必ず混在フォーマットを指定します。

混在フォーマット(変換前)

➡

混在フォーマット(変換後)

: 1つのレコード (1つの一般フォーマット)

: それぞれにフォーマット変換を定義

[A1]~[A4] : デザインシート(データ構造定義IDを示します)

この変換により、データ構造(データ型や桁数)が一般フォーマットとして扱える形になったため、以降の変換や格納などの処理では、このデータを一般フォーマットとして定義して扱うことができます。

- 混在フォーマットのデータを変換する場合、以下の変換・編集機能は利用できません。

- ー レコードの抽出機能
- ー データのチェック
- ー 項目編集およびレコード編集機能の一部

項目編集およびレコード編集機能は一部利用できません。利用可否は以下のとおりです。

項目編集	項目並び替え	○
	項目選択	○
	項目追加	○
	属性変換	○
	トリム	×
	パディング	×
	文字列の部分抽出	×
	文字列の設定	×
	文字列の連結	×
	大文字/小文字変換	×
	全角/半角変換	×
	ひらがな/カタカナ変換	×
	濁点・半濁点	×
	四則演算	×
	項目条件変換	×
	数値表記変換	×
レコード編集	日時の書式変換	×
	日時の部分取得	×
	振分け	×
	ソート	×
	連結(ジョイン・ユニオン)	×
	集計	×

○:利用可 ×:利用不可

変換が必要な場合は、混在フォーマットを他のフォーマット(一般、一般(繰返しあり)、伝票など)に変更して対処してください。

- 混在フォーマットで利用可能な文字コードは、以下のとおりです。

利用可能な文字コード	
Shift-JIS	
Unicode	UTF8
	UTF8-4
	CFSD
EUC	U90
	S90
JEF-EBCDIC	カナ

利用可能な文字コード	
	ASCII



## 付録E Information IntegratorがサポートするFTPのコマンド

FTPクライアントからInformation Integratorに接続する場合、使用できるFTPコマンドは以下のとおりです。

表E.1 Information Integratorで利用できるFTPコマンド

コマンド名	サポートの有 無	備考
USER	○	Information Integratorが受信できるUSER名は、最大20文字です。
PASS	○	Information Integratorが受信できるPASS名は、最大31文字です。
ACCT	×	—
CDUP	×	—
SMNT	×	—
REIN	×	—
QUIT	○	—
PORT	○	—
PASV	○(※1)	ASCII、IMAGEのみ対応
TYPE	○	—
STRU	×	—
MODE	△(※1)	どのモードが指定されても、ストリームモードとして動作します。
RETR	○	—
STOR	○	—
STOU	×	—
APPE	○	—
ALLO	×	—
REST	×	—
RNFR	×	—
RNTO	×	—
ABOR	○	—
DELE	×	—
RMD	×	—
CWD	○(※1) (※2)	Information Integratorは、転送ファイル名を絶対パスで扱うため、転送コマンド(STOR/RETR/APPE)で指定したファイル名とは、関係を持ちません。相対パスが通知された場合は、ユーザーのカレント配下にファイルを作成します。
MKD	○(※1) (※2)	Information Integratorの起動権限でディレクトリを作成します。
PWD	○(※1) (※2)	—
LIST	○(※1)(※2)	MS-DOS形式のみサポートします。オプション設定は無効となります。
NLST	×	—
SITE	○(※1)	“200 OK”の正常応答を行います。
STAT	×	—
SYST	○	—

コマンド名	サポートの有無	備考
HELP	×	—
NOOP	○(※1)	—

○:サポートしています。

△:条件によりサポートしています。詳細は備考を参照してください。

×:サポートしていません。

注1)ifitrf以外のサービス名を使用したパスでは、サポートしていません。

注2)ログイン利用者名のホームディレクトリではなく、Information Integratorの基準ディレクトリの情報を表示します。Information Integratorの基準ディレクトリは、以下のとおりです。

- ・ 配付先システムがWindowsの場合

Information Integratorサーバインストールディレクトリ
---------------------------------------

※ただし、転送コマンド実施後は前回転送時のファイル格納ディレクトリ

- ・ 配付先システムがUNIXの場合:

/ディレクトリ
---------

### Information IntegratorがサポートするFTPのサブコマンド

FTPサーバからInformation Integratorに接続する場合、使用できるサブコマンドは、put、get、append、typeおよびbyeです。

mget、mputコマンドなどは使用しないでください。

また、収集元システムがグローバルサーバ/PRIMEFORCEでTISPか、TISP/BPを使用している場合、ログインディレクトリ表示機能を使用しないでください。TISPおよびTISP/BPがFTPコマンドの機能として提供しているCENTER定義文のAUTOPWDオペランドに、必ず「NO」を指定してください。

### ファイル置換モードの対応

Information Integratorがファイル受信を行う場合、収集元システムで実行したコマンドに応じたファイル置換モードを実行します。

- ・ putサブコマンド: ファイル置き換えを行います。
- ・ appendサブコマンド: ファイル追加書きを行います。

## 付録F 変換・編集時に行うデータチェックについて

ここでは、Information Integratorが変換・編集時に行うデータチェックの内容について説明します。

### F.1 データ形式に対するチェック項目

以下の入力値チェックを行います。

- ・ 全角文字列属性
- ・ 日付・時間属性
- ・ 数値属性
- ・ バイナリ属性
- ・ ビット属性

以下にそれぞれのチェック内容について説明します。

#### F.1.1 全角文字列に対するデータチェック

以下のデータ型では、全角文字以外のデータを異常値として検出します。

- ・ SQL\_NCHAR
- ・ SQL\_VARNCHAR

#### F.1.2 日付・時間属性に対するデータチェック

以下のデータ型の異常を検知します。

- ・ 日付(SQL\_DATE)
- ・ 時間(SQL\_TIME)
- ・ 日付時間(SQL\_TIMESTAMP)

各データ型のチェック内容は、以下のとおりです。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値																
SQL_DATE	書式:[年][S][月][S][日] 年:1~9999以外、5桁以上 月:1~12以外、3桁以上 日:1~31以外、3桁以上 セパレータ:','または'-'以外	書式:ASCIIの場合 <table border="1"> <tr> <td>3</td><td>3</td><td>3</td><td>C</td><td>3</td><td>3</td><td>3</td><td>3</td> </tr> <tr> <td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td> </tr> </table> 3:3以外 N:0~9以外 C:3または4以外	3	3	3	C	3	3	3	3	N	N	N	N	N	N	N	N
3	3	3	C	3	3	3	3											
N	N	N	N	N	N	N	N											
SQL_TIME	書式:[時][S][分][S][秒] 時:0~23以外、3桁以上※2 分:0~59以外、3桁以上※2 秒:0~59以外、3桁以上※2 セパレータ:':'以外	書式:ASCIIの場合 <table border="1"> <tr> <td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td> </tr> <tr> <td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td> </tr> </table> 3:3以外 N:0~9以外	3	3	3	3	3	3	N	N	N	N	N	N				
3	3	3	3	3	3													
N	N	N	N	N	N													
SQL_TIMESTAMP	書式:[SQL_DATE書式][S][SQL_TIME書式] 日付部分:SQL_DATE書式に従います。	書式:ASCIIの場合 [SQL_DATE書式][SQL_TIME書式]																

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
	時間部分:SQL_TIME書式に従います。 セパレータ:','(空白)以外	日付部分:SQL_DATE書式に従います。 時間部分:SQL_TIME書式に従います。

### F.1.3 数値属性に対するデータチェック

以下のデータ型の異常を検知します。

- SQL\_INTEGER
- SQL\_SMALLINT
- SQL\_NUMERIC
- SQL\_DECIMAL
- SQL\_REAL
- SQL\_FLOAT

各データ型のチェック内容は、以下のとおりです。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値																																																								
SQL_INTEGER	書式:[符号部][整数部] 符号部: '+', '-', 省略以外 整数部: 0~9以外 入力値: -2147483648~2147483647以外	—																																																								
SQL_SMALLINT	書式:[符号部][整数部] 符号部: '+', '-', 省略以外 整数部: 0~9以外 入力値: -32768~32767以外	—																																																								
SQL_NUMERIC	書式:[符号部][整数部][.][小数部] 符号部: ('+', '-', 省略)以外 ※ 整数部・小数部: 0~9以外 ※ 符号有無「なし」の場合は符号部を省略してください。また、正の値の場合は符号を省略可能です。	書式: 符号位置が先頭「ビット表現」の場合 <table border="1" style="margin-left: 20px;"> <tr><td>C</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td></tr> </table> 符号位置が先頭「文字表現」の場合 <table border="1" style="margin-left: 20px;"> <tr><td>符</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td></tr> <tr><td>号</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td></tr> </table> 符号位置が末尾「ビット表現」の場合 <table border="1" style="margin-left: 20px;"> <tr><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>C</td></tr> <tr><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td></tr> </table> 符号位置が末尾「文字表現」の場合 <table border="1" style="margin-left: 20px;"> <tr><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>F</td><td>符</td></tr> <tr><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>号</td></tr> </table> N: 0~9以外 符号: 文字表現 '+', '-' 以外 JEFコード系	C	F	F	F	F	F	F	N	N	N	N	N	N	N	符	F	F	F	F	F	F	号	N	N	N	N	N	N	F	F	F	F	F	F	C	N	N	N	N	N	N	N	F	F	F	F	F	F	符	N	N	N	N	N	N	号
C	F	F	F	F	F	F																																																				
N	N	N	N	N	N	N																																																				
符	F	F	F	F	F	F																																																				
号	N	N	N	N	N	N																																																				
F	F	F	F	F	F	C																																																				
N	N	N	N	N	N	N																																																				
F	F	F	F	F	F	符																																																				
N	N	N	N	N	N	号																																																				

データ型	CSVファイルでの異常値	バイナリファイルでの異常値												
		C: a,b,c,d,e,f以外 F:F以外 JEFコード系以外 C:3,4,5,7以外 F:3以外												
SQL_DECIMAL	書式:[符号部][整数部][.][小数部] 符号部:('+', '-', 省略)以外 ※ 整数部・小数部:0~9以外 ※ 正の値の場合は符号を省略可能です。	書式:符号位置の指定はできません。 <table border="1" style="margin: 10px auto;"> <tr> <td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>N</td> </tr> <tr> <td>N</td><td>N</td><td>N</td><td>N</td><td>N</td><td>C</td> </tr> </table> N:0~9以外 C:(a,b,c,d,e,f)以外	N	N	N	N	N	N	N	N	N	N	N	C
N	N	N	N	N	N									
N	N	N	N	N	C									
SQL_REAL	書式:[符号部1][整数部][.][小数部][E][符号部2][指数部]※ 符号部1:('+', '-', 省略)以外 符号部2:('+', '-')以外、省略 整数部、小数部、指数部:0~9以外 E:('E', 'e')以外 範囲:IEEE 754 形式単精度浮動小数点数の値域以外 ※ [E][符号部2][指数部]は省略可能です。省略する場合はすべて省略してください。	—												
SQL_FLOAT	書式:[符号部1][整数部][.][小数部][E][符号部2][指数部] ※ 符号部1:('+', '-', 省略)以外 符号部2:('+', '-')以外、省略 整数部、小数部、指数部:0~9以外 E:('E', 'e')以外 範囲:IEEE 754 形式倍精度浮動小数点数の値域以外 ※ [E][符号部2][指数部]は省略可能です。省略する場合はすべて省略してください。	—												

## F.1.4 バイナリ属性に対するデータチェック

以下のデータ型の異常を検知します。

- SQL\_BINARY
- SQL\_LONGVARBINARY

各データ型のチェック内容は、以下のとおりです。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
SQL_BINARY	BASE64で利用不可能な文字	—
SQL_LONGVARBINARY	BASE64で利用不可能な文字	—





## 参考

BASE64で利用可能な文字は、以下の半角文字です。

アルファベット大文字	A～Z
アルファベット小文字	a～z
数字	0～9
記号	‘+’、‘/’、‘=’

## F.1.5 ビット属性に対するデータチェック

以下のデータ型の異常を検知します。

- SQL\_BIT

各データ型のチェック内容は、以下のとおりです。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
SQL_BIT	0(半角)または1(半角)以外の文字	—

## F.2 桁数の過不足に対するデータチェック

### 桁数あふれのチェック

入力データの桁数が定義よりも多い場合、異常値とします。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
SQL_CHAR	入力データの桁数 > 定義の桁数 整数部、小数部がある場合は整数部の桁数・小数部の桁数を別々にチェックします。	—
SQL_NCHAR		
SQL_DECIMAL		
SQL_NUMERIC		
SQL_BINARY		
SQL_BIT		
SQL_VARCHAR		データ長領域の値※ > 定義の桁数
SQL_VARNCHAR		
SQL_LONGVARBINARY		

※ データ長領域とは、SQL\_VARCHAR、SQL\_VARNCHAR、SQL\_LONGVARBINARYをバイナリで表現している場合に、データの一部として自身のデータ長を格納している領域です。Information Integratorでは、その領域に格納されているデータ長と定義の桁数とを比較することにより桁数のあふれをチェックします。

### 桁数不足のチェック

入力データの桁数が定義よりも少ない場合、異常値とします。

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
SQL_CHAR	入力データの桁数 < 定義の桁数 整数部、小数部がある場合は整数部の桁数・小数部の桁数を別々にチェックします。	—
SQL_NCHAR		
SQL_DECIMAL※		

データ型	CSVファイルでの異常値	バイナリファイルでの異常値
SQL_NUMERIC※		
SQL_BINARY		
SQL_BIT		

※ 変換前データに対してのみチェックできます。

## F.3 必須入力データのチェック

---

必須入力データの有無をチェックします。

### F.3.1 NULL値のチェック

---

すべてのデータ属性に対して、入力値がNULLでないかのチェックを実施します。

CSVファイルでは、入力データの区切り文字‘,’の間に何も無い状態をNULLとして、NULLの場合に異常値とします。

バイナリファイルでは、null制約が有効な場合に、データにnull識別子が存在する状態をNULLとして、NULLの場合に異常値とします。null制約についての詳細は、“III Studio ヘルプ”または、“デザインシート操作ヘルプ”を参照してください。

## 付録G ファイアウォール環境への導入時の注意事項

ここでは、Information Integratorを、ファイアウォール環境へ導入する場合の注意事項を説明しています。

### G.1 FTP/FTP+/FTPMでのコネクション確立シーケンス(前提知識)

FTP、FTP+、またはFTPMは、以下の2つのコネクションを利用してファイル転送を実現しています。

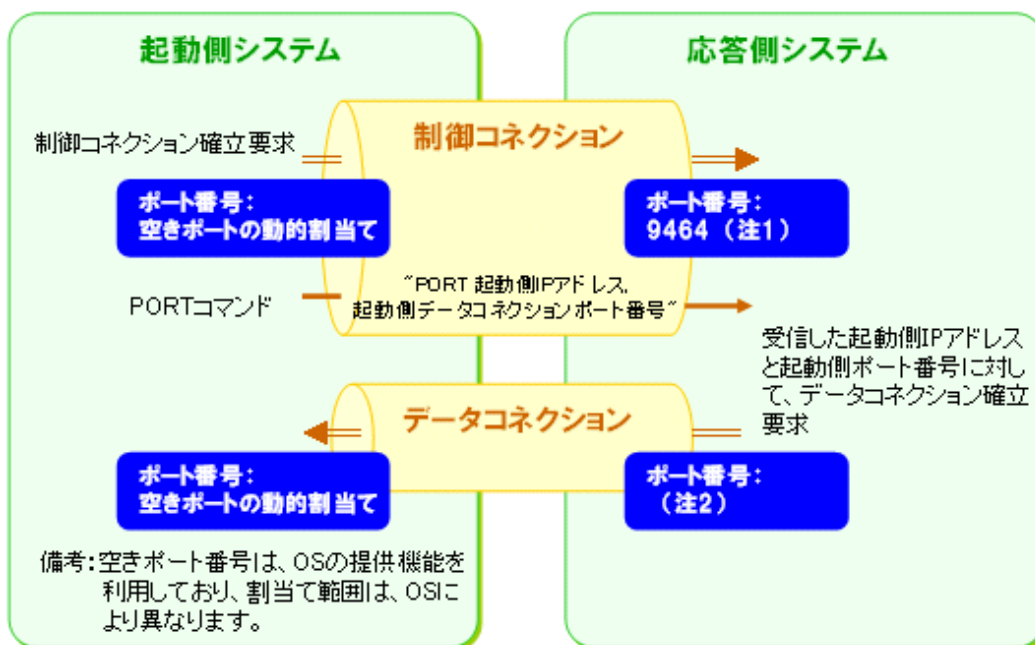
- ・ 制御コネクション(FTPプロトコル電文を受け渡しするコネクションです)
- ・ データコネクション(ファイル転送データを受け渡しするコネクションです)

ファイアウォールは、これらのコネクション確立の際にポートフィルタリングなどの制御を実施します。このため、前提知識として、FTP、FTP+、またはFTPMにおけるコネクション確立シーケンスを以下の図に示します。

なお、FTP、FTP+、またはFTPMでは、以下の2つのモードがあり、データコネクション確立方向が異なります。

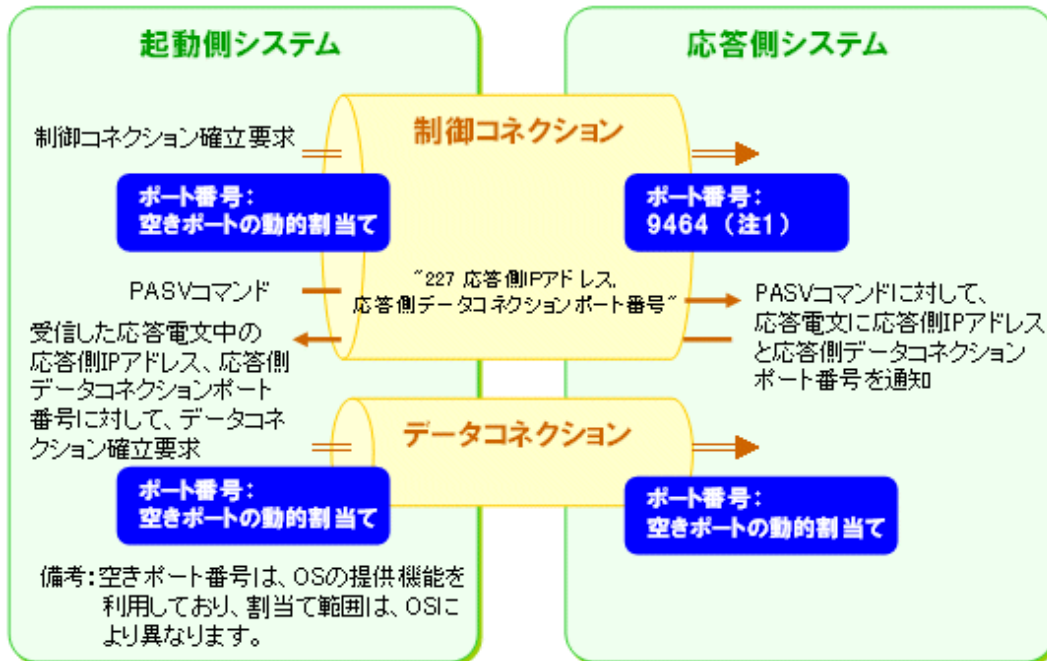
- ・ PORTモード
- ・ PASVモード

図G.1 PORTモード時のコネクション確立シーケンス



- ・ 注1:  
Information Integratorサーバは、servicesファイル内の「ifitnrf」の指定値で待ち受けします。なお、起動側から指定する応答側制御コネクションポート番号設定箇所は、データソース定義の「ポート番号」です。  
なお、標準FTPサーバの場合、応答側制御コネクションポート番号は、「21」です。
- ・ 注2:  
動作環境ファイルの「dport」の指定値です。動作環境ファイルについては、“セットアップガイド”を参照してください。

図G.2 PASVモード時のコネクション確立シーケンス



・ 注1:

Information Integratorサーバは、servicesファイル内の「ifitnrf」の指定値で待ち受けします。なお、起動側から指定する応答側制御コネクションポート番号設定箇所は、データソース定義の「ポート番号」です。

なお、標準FTPサーバの場合、応答側制御コネクションポート番号は、「21」です。

## G.2 ファイアウォール環境での問題発生箇所と対処

ファイアウォール環境では、ファイアウォール側の設定により、以下の問題が発生する可能性があります。このため、Information Integratorサーバのデータ転送に合わせて、ファイアウォール側の設定を変更する必要があります。

### G.2.1 フィルタリング

#### [問題発生箇所]

ファイアウォール側の設定(ポート番号のフィルタリングなど)により、コネクション確立が失敗します。FTP、FTP+、またはFTPMで、PORTモードの場合は、双方向でコネクション確立を実施するため、注意が必要です。

#### [対処]

ファイアウォール機器によりフィルタリング設定方法は異なるため、利用しているファイアウォール機器の設定方法を確認してください。また、運用システムのセキュリティポリシーに従った対処を実施してください。

また、Windows Server 2003 Service Pack 1以降およびWindows XP Service Pack 2以降を適用しているシステムで、Windowsファイアウォールを「有効」としている場合、以下の手順でInformation Integratorサーバを「通信を許可するプログラム」としてシステムへ登録してください。

1. コントロールパネル内の「Windowsファイアウォール」を実行します。設定パネル内の[例外を許可しない]にチェックが入っていないことを確認します。
2. [例外]タブを選択して、[プログラムの追加]ボタンをクリックします。以下のファイルを登録します。
  - Information Integratorサーバインストールディレクトリ¥bin¥f3jndefsv.exe
  - Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jnsdisp.exe
  - Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntrnintdm.exe
  - Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntrnrspdm.exe

- Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntnrnsphcs.exe
- Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntnrnvchcs.exe
- Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntnrntccntl.exe
- Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntnrnintftp.exe
- Information Integratorサーバインストールディレクトリ¥lnk¥bin¥f3jntnrnrspftp.exe

3. [例外]タブにて上記ファイル名の左にチェックが入っているのを確認して、設定パネルの[OK]ボタンをクリックします。

## G.2.2 NAT変換

### [問題発生箇所]

NAT変換(プライベートアドレスとグローバルアドレスの変換をルータなどの機器で実現)を実施されている場合、データコネクション確立処理に失敗する可能性があります。これは、制御コネクション上に流通するFTPプロトコル電文をNAT変換機器が正しくNAT変換しない場合、問題となります。

FTPプロトコル電文として、PORTコマンドまたはPASVコマンド電文でIPアドレス情報が流通します。NAT変換機器で「FTP転送を許可」とした場合、NAT変換機器は応答側制御コネクションポート番号が「21」の時のみFTPプロトコル電文上のIPアドレスをNAT変換します。Information Integratorサーバの応答側制御コネクションポート番号の標準は、「9464」のため、PORTコマンドまたはPASVコマンド電文のIPアドレスがNAT変換されず、データコネクション確立処理が失敗します。

### [対処]

NAT変換機器において、応答側制御コネクションポート番号が「21」以外の場合でもNAT変換可能なように設定してください。なお、NAT変換機器により設定方法は異なりますので、NAT変換機器の設定方法を確認してください。「21」以外の場合、NAT変換できない機器がありますので、ご注意ください。

また、応答側制御コネクションポート番号を標準FTPと同じ「21」で設定することでNAT変換の問題を解決することができます。変更手順を以下に説明します。なお、「21」に変更する場合のデメリットとしては、標準FTPサーバとInformation Integratorサーバとの共存ができません。

#### 起動側システムの設定

1. 相手側ポート番号に「21」を設定します。
  - Information Integratorサーバの場合: データソース定義の「ポート番号」に「21」を指定します。
  - グローバルサーバ/PRIMEFORCEの場合: ネットワーク定義の該当するKGFGNPTH定義命令のportオペランドに「21」を指定します。
2. 定義を登録します。
  - Information Integratorサーバの場合: III Studioまたは、デザインシートから定義を登録します。
  - グローバルサーバ/PRIMEFORCEの場合: 定義作成JCLを実行します。

#### 応答側システムの設定

1. servicesファイルの設定を変更します。

変更前:

ftp	21/tcp
ifitrnf	9464/tcp

変更後:

#ftp	21/tcp	#コメント行にします
#ifitrnf	9464/tcp	#9464情報を退避し、
ifitrnf	21/tcp	#21へ変更します

2. 標準FTPサーバを停止します。

3. 動作環境ファイルの「dport」に「20」を設定します。動作環境ファイルについては、“セットアップガイド”を参照してください。
4. Information Integratorサーバを再起動します。Information Integratorサーバの再起動方法については、“セットアップガイド”を参照してください。

## 付録H 組み込み可能なアプリケーション

組み込み可能なアプリケーションは、以下のとおりです。

### H.1 処理プラグイン機能で組み込み可能なアプリケーション

#### 入出力パラメタ

入出力パラメタは、以下のいずれかの方法で指定します。

- ・「-t」、「-bs」などの固定のパラメタ値
- ・ 動的パラメタ値を指定

利用可能な動的パラメタは、以下のとおりです。

表H.1 指定可能な動的パラメタ

動的パラメタ	名称	説明
\$IFI_InputFile	入力データファイル名	入力データとなるファイル名が通知されます。
\$IFI_OutputFile	出力データファイル名	処理データを出力するファイル名が通知されます。

#### 注意

- プラグイン定義に動的パラメタ(「\$IFI\_InputFile」および「\$IFI\_OutputFile」)を指定する場合は、1つのプラグイン定義内で、「\$IFI\_InputFile」および「\$IFI\_OutputFile」を両方指定する必要があります。
- データ収集ファンクションの前に、プラグインのファンクションを指定する場合、およびデータ配付ファンクションの後に、プラグインのファンクションを指定する場合、「\$IFI\_InputFile」および「\$IFI\_OutputFile」は指定できません。
- プラグイン定義に、動的パラメタ「\$IFI\_OutputFile」を指定する場合、利用者のプラグインで、「\$IFI\_OutputFile」で通知されたファイルを必ず作成してください。
- プラグイン定義に動的パラメタ(「\$IFI\_InputFile」および「\$IFI\_OutputFile」)を指定する場合、Information Integratorが通知するファイルには半角空白が含まれます。したがって、引数を"(ダブルクォート)で囲むなど、必ず利用者のプラグインで空白が認識できるようにしてください。

#### 環境変数

以下の環境変数を設定できます。

- ・ 利用者独自の環境変数
- ・ Information Integratorの環境変数(EXE形式のプラグインのみ)

Information Integratorのさまざまな実行情報をアプリケーションに通知できます。

EXE形式のプラグインで利用できるInformation Integratorの環境変数は、以下のとおりです。

項番	環境変数名	設定値	利用例
1	IFI_FunctionID	実行中のファンクションID(プラグインID)が設定されます。	設定された値を元に、定義IDごとに処理を振り分けられるアプリケーションの作成
2	IFI_FuncsvcID	実行中のファンクションの処理番号(1~999999999)が通知されます。	通知された値をファイル名に埋め込むことで、プラグインで出力するファイル名を一意にする

項番	環境変数名	設定値	利用例
		なお、通知される処理番号は、システムで一意です。同じファンクションが同時に実行されても、必ず異なる番号が通知されます。	
3	IFI_ProcID	プラグインが実行されたプロセスの定義IDが設定されます。	設定された値を元に、定義IDごとに処理を振り分けられるアプリケーションの作成
4	IFI_SvcID	実行中のプロセスのサービス番号が通知されます。 なお、通知されるサービス番号は、プロセスごとに異なります。同じプロセスが同時に実行されても、必ず異なる番号が通知されます。ただし、同じプロセス内で定義されたファンクションに対しては、同じサービス番号が通知されます。	本環境変数により通知された値や、「IFI_ProcID」などで設定された値などを元に、Information Integratorの状態表示コマンド(ifidisp)を実行。 これにより、他のファンクションの情報や、プラグイン自身の通知されていない情報の取得を実施。  状態表示コマンドについての詳細は、“コマンドリファレンス”を参照してください。



## 注意

### 利用者独自の環境変数を利用する場合

[IFI\_]から始まる環境変数名を指定することはできません。

### 処理完了の待ち合わせについて

Information Integratorサーバでは、以下のいずれかの処理形態をとります。

- アプリケーションの処理完了を待ち合わせてから、以降の処理を行う  
アプリケーションの処理結果も含めて管理することが可能です。
- アプリケーションの処理完了を待ち合わせずに、以降の処理を行う  
アプリケーションの処理を実行したかどうかのみ管理します。

アプリケーションの処理完了を待ち合わせてから、以降の処理を行う形態の場合、アプリケーション内に、以下の指定方法で復帰値を返却します。

なお、判定の基準となる「終了コード」は、プラグイン定義に設定します。

### 指定可能な復帰値

指定方法	説明
すべて	常に正常と判断します。
= 終了コード	指定された終了コードの場合に、正常終了したとみなします。
> 終了コード	指定された終了コードより大きい場合に正常終了とみなします。
< 終了コード	指定された終了コードより小さい場合に正常終了とみなします。

### アプリケーションの利用権限について

アプリケーションを格納するディレクトリおよびアプリケーションは、Information Integrator特権ユーザーがアクセス可能な状態にしてください。



## 標準出力、標準エラー出力への文字列出力について

アプリケーションから標準出力、標準エラー出力に文字列を出力すると、処理プラグインログに当該文字列を埋め込んで出力することができます。これにより、プラグインの処理に関する情報を出力することができます。



### 注意

プラグインの処理結果を待ち合わせて後続の処理プロセスを実行する場合のみ、利用可能です。プラグイン定義の終了確認の指定値は、必ず「同期」を指定してください。

- 処理プラグインログのメッセージ出力契機

以下の契機のいずれかとなります。出力しない設定となっている場合は、本ログは出力されません。

- プラグインが異常終了したとき(初期値)
- プラグイン実行時は、正常、異常に関わらず常に出力する
- 常に出力しない

これらは、動作環境ファイルの「plglog\_putmsg」に指定した値に準じます。

- 処理プラグインログの出力最大サイズ

32Kバイトまでの情報を出力することができます。32Kバイトを越える情報が出力された場合は、切り捨てられます。

- 処理プラグインログの出力場所

- 運用管理クライアントの業務履歴に関するプロパティの[処理番号]をクリックすると表示されます。詳細は、“運用管理クライアント ヘルプ”を参照してください。
- 異常時に出力されるメッセージ(「IF13305」または「IF23305」)のパラメタ「msgs」パラメタに出力されます。



### 注意

メッセージ本文への出力では、512バイトまでの文字列のみ出力されます。

運用管理クライアントへの表示では、32Kバイトまで表示します。512バイトを越える文字列が出力されたときは、運用管理クライアントで確認してください。

## H.2 変換プラグイン機能で組み込み可能なアプリケーション

### H.2.1 アプリケーションの開発環境

- C言語の場合

- Windowsの場合

Microsoft® Visual Studio 2005

- Solarisの場合

Sun WorkShop 4.0以降

- Linuxの場合

GNU C Compiler

- COBOL言語の場合

Fujitsu NetCOBOL 7.0以降

## 注意

- スレッドについて

Information Integratorからはマルチスレッド構成として実行されるため、アプリケーション内ではスレッドセーフな関数(マルチスレッドモジュール内で実行しても安全な関数)を使用してください。

- Windowsのコンパイルオプションについて

Windowsでの利用時の、Microsoft® Visual Studioでコンパイルする際には、“/MDオプション”(ランタイムライブラリにマルチスレッドDLLを使用する)を設定してください。

## H.2.2 アプリケーションの利用権限

アプリケーションの動的リンクライブラリに必要なアクセス権は、以下のとおりです。

- Information Integrator運用ユーザーに対して読み取り権を付与してください。
- Information Integrator運用ユーザー単位でアクセスできるように設計してください。

アクセス権が不足すると、アプリケーション実行時にエラーとなります。

## H.2.3 利用可能なアプリケーションインタフェース

- 復帰値の型

プラグインの処理が正常に完了したか否かの判定時に利用します。

- 関数名

APIやメソッド名など呼び出すアプリケーション名を指定します。

- 引数

Information Integratorと、アプリケーション間での情報の受け渡しに必要な引数を指定します。1個～31個までの任意の引数を指定できます。

引数ごとに、以下のような入出力時のデータの型やデータの流れなどをアプリケーションの引数に指定できます。

- 入出力時のデータの型(整数型や文字列型、内部10進数など)。例えば、引数の型(long型やchar\*型など)
- Information Integratorからアプリケーションに渡される入力データ
- アプリケーションが処理した結果を、アプリケーションからInformation Integratorに返却するデータ

## ポイント

アプリケーションに対して以下のような情報を引数で受け渡すこともできます。

- Information Integratorの実行情報

例えば、実行情報を元に実行日時をデータ項目に結合するなどの独自のデータ変換やデータ整形を行うことができます。詳細は、“[H.2.9 実行情報の受け渡し](#)”を参照してください。

- 固定文字列

- エラー発生時に利用する独自のエラーメッセージ(100バイト以内の英数字文字列)

- 形式

以下のような形式で宣言できるインタフェースが利用可能です。

```
復帰値の型 関数名(引数1,引数2,・・・引数31);
```

関数例)

```
int user_func1 (char *param1, long param2, char *result);
```

## H.2.4 アプリケーションで利用可能なデータの文字コード

---

- Shift-JIS
- EUC
  - U90
  - S90
- Unicode
  - UTF8
  - UTF8-4



### 注意

アプリケーションへ受け渡すデータの文字コードについて

Information Integratorからアプリケーションへ受け渡すデータの文字コードは、アプリケーションの入力データとして利用する文字コードと必ず一致させてください。

## H.2.5 利用可能な引数の型

---

利用可能な引数の型は、以下のとおりです。

### C言語の場合

- int(32ビット環境のみ利用可能。64ビット環境の場合は、「long」を利用してください。)
- long
- longlong(\_int64)(64ビット環境のみ利用可能です。なお、「\_int64」は、Windows環境での表現です。)
- char\*
- double\*
- float\*
- int\*
- long\*
- longlong\*

### COBOL言語の場合

- PIC 9(4)(Information IntegratorがWindows環境のみ利用可能)
- PIC 9(8)(Information IntegratorがUNIX64ビット環境のみ利用可能)
- USAGE COMP-1
- PIC X(n)(n:2以上の正の整数)

## H.2.6 利用可能なデータ型

---

利用可能なデータ型は、以下のとおりです。

- データ形式がCSVファイル形式の場合  
すべてのデータは「SQL\_VARCHAR」として扱われます。

- データ形式がバイナリファイル形式の場合

以下のデータ型が利用可能です。

- SQL\_VARCHAR(可変長英数字文字列)

有効データを特定できるものに限ります。C言語の場合は、必ずNull終端を保証してください。

- SQL\_INTEGER(整数)

- SQL\_NUMERIC(外部10進数属性)

- SQL\_DECIMAL(内部10進数属性)

- SQL\_FLOAT(倍精度浮動小数点属性)

上記3つのデータ型では、いずれも受渡しデータは文字列として扱います。

- SQL\_DATE(日付属性)

- SQL\_TIME(時間属性)

- SQL\_TIMESTAMP(日付時間属性)

上記3つのデータ型では、いずれもプラグイン実行後の形式が、「YYYY-MM-DD hh:mm:ss」表記となります。

## H.2.7 引数の型とデータ型の関係

利用可能な引数の型とデータ型の対応関係は、以下のとおりです。

### C言語の場合

- CSVファイル形式の場合

利用可能な引数の型とデータ型は以下のとおりです。

項番	引数型	データ型	
1	char*	SQL_VARCHAR	可変長英数字文字列

- バイナリファイル形式の場合

利用可能な引数の型とデータ型は以下のとおりです。

項番	引数型	データ型	
1	int	SQL_INTEGER	整数
2	long	SQL_INTEGER	整数
3	longlong(_int64)	SQL_INTEGER	整数
4	char*	SQL_VARCHAR	可変長英数字文字列
		SQL_NUMERIC	外部10進数属性
		SQL_DECIMAL	内部10進数属性
		SQL_DATE	日付属性
		SQL_TIME	時間属性
5	double*	SQL_FLOAT	倍精度浮動小数点属性
		SQL_FLOAT	倍精度浮動小数点属性
6	float*	SQL_FLOAT	倍精度浮動小数点属性

項番	引数型	データ型	
7	int*	SQL_INTEGER	整数
8	long*	SQL_INTEGER	整数
9	longlong*	SQL_INTEGER	整数

## COBOL言語の場合

- CSVファイル形式の場合

利用可能な引数の型とデータ型は以下のとおりです。

項番	引数型	データ型	
1	PIC X(n) (n:2以上の正の整数)	SQL_VARCHAR	可変長英数字文字列

- バイナリファイル形式の場合

利用可能な引数の型とデータ型は以下のとおりです。

項番	引数型	データ型	
1	PIC 9(4) PIC 9(8)	SQL_INTEGER	整数
2	USAGE COMP-1	SQL_FLOAT	倍精度浮動小数点属性
3	PIC X(n) (n:2以上の正の整数)	SQL_VARCHAR	可変長英数字文字列
		SQL_NUMERIC	外部10進数属性
		SQL_DECIMAL	内部10進数属性
		SQL_DATE	日付属性
		SQL_TIME	時間属性
	SQL_TIMESTAMP	日付時間属性	

## H.2.8 復帰値の型

アプリケーションの復帰値の属性および、各属性での、正常／異常の判断方法は以下のとおりです。

項番	復帰値の属性	アプリケーションの正常／異常の判断
1	int	プラグイン定義の「終了コードの正常異常判断基準」に従って判断します。詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。
2	long	
3	void(復帰値無し)	正常／異常の判断は行いません。
4	文字列型(char*)	プラグイン定義の「終了コードの正常異常判断基準」に従って判断します。詳細は、「III Studio ヘルプ」または、「デザインシート操作ヘルプ」を参照してください。



処理結果のデータを復帰値で返すようなアプリケーションを復帰値に指定することはできません。

## H.2.9 実行情報の受け渡し

アプリケーションに対して実行情報を入力データとして受け渡します。これにより、実行情報を元に実行日時をデータ項目に結合するなどの独自のデータ変換やデータ整形を行うことができます。

アプリケーションへ受け渡すことができる実行情報は、以下のとおりです。

項番	指定内容	指定形式	備考
1	サービス番号	@IFI_SvcID@	SQL_VARCHARとして受け渡します。
2	実行日時	@IFI_SysDate@	システム日付を「YYYYMMDD」の形式で通知します。 YYYY: 西暦 MM: 月 DD: 日  SQL_VARCHARとして受け渡します。
3	処理番号	@IFI_FuncsvcID@	SQL_VARCHARまたは、SQL_INTEGERとして受け渡します。
4	ファンクションID	@IFI_FunctionID@	SQL_VARCHARとして受け渡します。
5	プロセスID	@IFI_ProcID@	SQL_VARCHARとして受け渡します。

## H.2.10 アプリケーションの呼び出し方法

アプリケーションは、以下のようにインタフェースに関する定義と実行に関する定義をそれぞれ以下の定義が必要です。

定義パラメタの詳細については、“III Studio ヘルプ”または、“デザインシート操作ヘルプ”を参照してください。

- ・ インタフェースに関する定義
  - ー プラグイン定義  
アプリケーションのインタフェースを指定します。
  - ー フォーマット定義  
文字コードやデータ形式等のデータに関する内容を指定します。
- ・ アプリケーションの実行に関する定義
  - ー データ変換定義  
アプリケーションの実行に関する内容を指定します。

### ポイント

#### 各定義の入力データの関係について

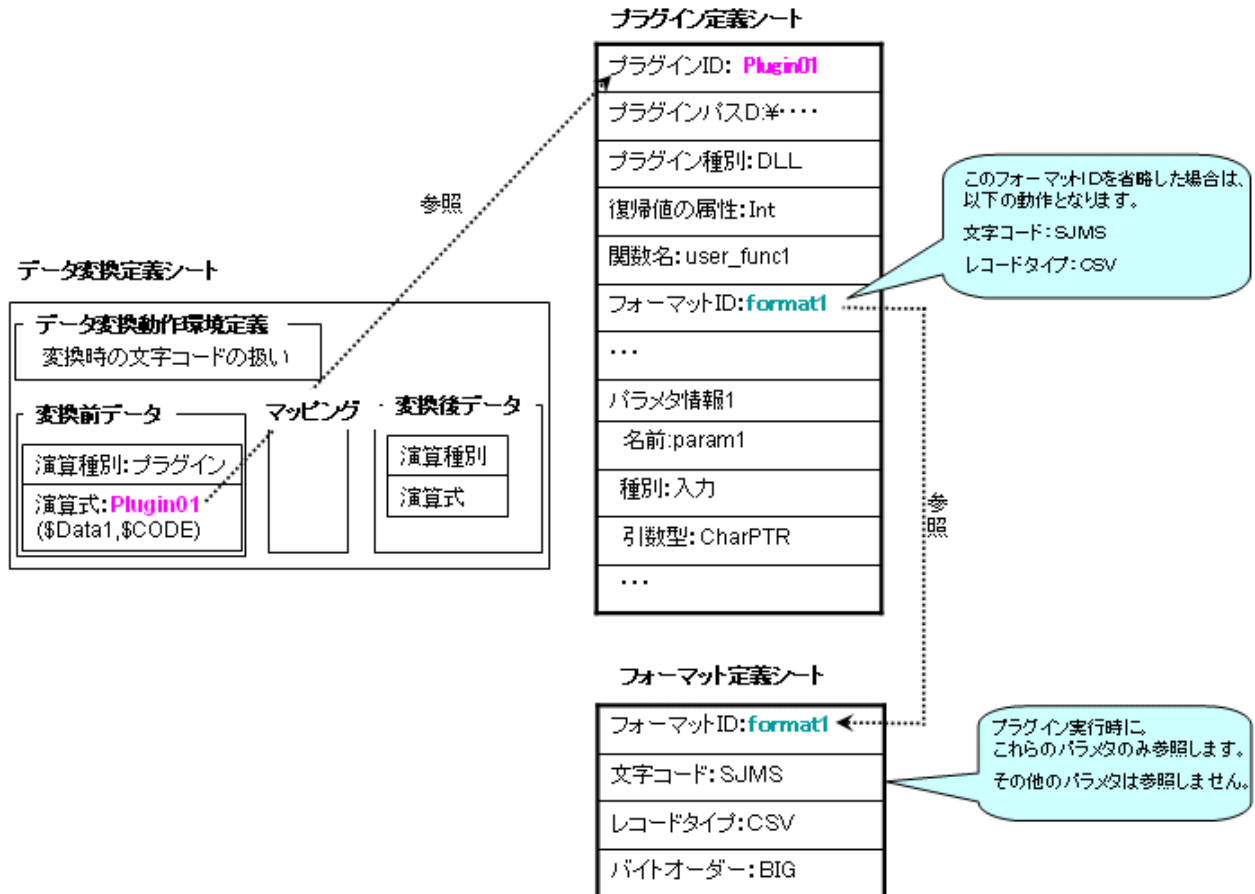
- ・ プラグインID  
プラグイン定義で入力する「プラグインID」は、データ変換定義の変換対象となるデータ構造の演算式に指定します。
- ・ パラメタ情報  
データ変換定義で入力する「パラメタ情報」は、プラグイン定義で入力する「パラメタ情報」の指定順序に合わせて、項目名や固定文字列などを指定します。  
  
なお、データ構造定義の項目を入力データとして使用する場合は「\$項目名」で指定します。固定文字列は“(ダブルクォート)”で囲んで指定します。

- 文字コード

データ変換時にInformation Integratorの内部で扱うデータの文字コードは、「変換時の文字コードの扱い」に指定した文字コードです。

アプリケーションで扱うデータの文字コードは、プラグイン定義で指定したフォーマット定義に依存します。

デザインシートのデータ変換定義シートとプラグイン定義シートの関係は、以下のとおりです。



## H.2.11 アプリケーションとプラグイン定義の例

C言語のアプリケーションを例に、アプリケーションの作成例、プラグイン定義の定義例およびアプリケーション実行時の各パラメタのデータ例を示します。

### 例1

アプリケーションの例

関数名: copy\_func

復帰値の型: char\*

関数仕様: 引数「in\_data」に指定した入力データを引数「out\_data」に指定した出力バッファに引数「out\_size」に指定したサイズの数だけコピーします。

```
char *copy_func(
    char *out_data, /* 結果出力 */
    char *in_data, /* 入力 */
    long out_size /* 結果出力用領域サイズ */
)
```

```

{
    /*-----*
    * 結果出力“out_data”に、領域サイズ“out_size”をオーバ-
    * しない範囲で入力“in_data”をコピーします。
    *-----*/
    strncpy(out_data, in_data, out_size);
    return out_data; /* 復帰 */
}

```

### プラグイン定義の例

プラグイン定義には、アプリケーション例の太字部分の情報を定義します。各パラメタに関する指定値について説明します。

- 第1パラメタ名「out\_data」

本パラメタは結果を出力するパラメタです。そのため、定義の「種別」には、「結果」を指定します。

項目名(パラメタ名)	指定内容
名前	out_data
種別	結果
引数型	CharPTR
データ型	SQL_VARCHAR

- 第2パラメタ名「in\_data」

本パラメタは入力を表わすパラメタです。そのため、定義の「種別」には、「入力」を指定します。

項目名(パラメタ名)	指定内容
名前	in_data
種別	入力
引数型	CharPTR
データ型	SQL_VARCHAR

- 第3パラメタ名「out\_size」

本パラメタは実際のデータには含まれない引数ですが、定義の「種別」に「出力データサイズ」と指定することにより、アプリケーションを呼び出すことができます。

項目名(パラメタ名)	指定内容
名前	out_size
種別	出力データサイズ
引数型	Long
参照先項目	out_data

### データ例

上記のプラグインを実行したときの各パラメタのデータ例です。パラメタ「in\_data」に「apple」を入力した場合、以下のとおり、結果データは入力データと同じになります。

入力データ	結果データ
apple	apple

## 例2

### アプリケーションの例

関数名: setstring\_func



復帰値の型:int

関数仕様:引数「out\_format」に指定した書式に、引数「code\_no」に指定した入力データと引数「in\_data」に指定した入力データを埋め込みます。埋め込んだデータは、引数「out\_data」に指定した出力バッファに出力します。

```
int setstring_func(
    char *out_data, /* 結果出力 */
    char *out_format, /* 出力の形式 */
    long code_no, /* 入力 (数値) */
    char *in_data /* 入力 (文字列) */
)
{
    int ret;
    /*-----*
    * 出力の形式"out_format"に指定した形で、入力"code_no"、入力"in_data"
    * を組み合わせて、結果出力"out_data"に出力します。
    *-----*/
    ret = sprintf(out_data, out_format, code_no, in_data);
    return ret; /* 復帰 */
}
```

### プラグイン定義の例

プラグイン定義には、アプリケーション例の太字部分の情報を定義します。各パラメタに関する指定値について説明します。

- 第1パラメタ名「out\_data」

項目名(パラメタ名)	指定内容
名前	out_data
種別	結果
引数型	CharPTR
データ型	SQL_VARCHAR

- 第2パラメタ名「out\_format」

アプリケーション自身の入力データとなりますが、常に固定とする出力の形式を指定するため、種別を「デフォルト」と定義します。種別に「デフォルト」を定義した場合は、あわせて「デフォルト設定値」への設定が必要です。出力の形式を指定します。

項目名(パラメタ名)	指定内容
名前	out_format
種別	デフォルト
引数型	CharPTR
デフォルト設定値	No(%d) Data(%s) この例では、書式化文字列を指定します。データ変換定義に演算式指定するときの引数情報(項目)を省略することができます。
データ型	SQL_VARCHAR

- 第3パラメタ名「code\_no」

項目名(パラメタ名)	指定内容
名前	code_no
種別	入力
引数型	Long

項目名(パラメタ名)	指定内容
参照先項目	SQL_INTEGER

- 第4パラメタ名「in\_data」

項目名(パラメタ名)	指定内容
名前	in_data
種別	入力
引数型	CharPTR
データ型	SQL_VARCHAR

#### データ例

上記のプラグインを実行したときの各パラメタのデータ例です。各パラメタのデータは、以下のようになります。

パラメタ	定義の「種別」に指定した値	データ例
in_data	入力	KOBE
code_no	入力	1000
out_data	結果	No.(1000) Data(KOBE)
out_format	デフォルト	No(%d) Data(%s)

### 例3

#### アプリケーションの例

関数名: cutstring\_func

復帰値の型: int

関数仕様: 引数「in\_data」に指定した入力データを引数「in\_string」に指定した書式にします。そのデータを、以下の各引数に指定した出力バッファにそれぞれ出力します。

- 引数「result」
- 引数「out\_data1」
- 引数「out\_data2」

このうち、実行結果として返却するのは 引数「result」の値となります。

```
int cutstring_func(
    char *in_data, /* 入力 (文字列) */
    char *in_string, /* 入力 (書式化文字列) */
    char *result, /* 結果出力 */
    char *outdata1, /* 出力 (文字列) */
    char *outdata2 /* 出力 (文字列) */
)
{
    int ret;
    /*-----*
    * 入力"in_data"を入力形式"in_string"に指定した形で、結果出力"result"、
    * 出力"outdata1"、出力"outdata2"に出力します。
    *-----*/
    ret = sscanf(in_data, in_string, result, outdata1, outdata2);
    return ret; /* 復帰 */
}
```

#### プラグイン定義の例

プラグイン定義には、アプリケーション例の太字部分の情報を定義します。各パラメタに関する指定値について説明します。

- 第1パラメタ名「in\_data」

項目名(パラメタ名)	指定内容
名前	in_data
種別	入力
引数型	CharPTR
データ型	SQL_VARCHAR

- 第2パラメタ名「in\_string」

項目名(パラメタ名)	指定内容
名前	in_string
種別	入力
引数型	CharPTR
データ型	SQL_VARCHAR

- 第3パラメタ名「result」

項目名(パラメタ名)	指定内容
名前	result
種別	結果
引数型	CharPTR
データ型	SQL_INTEGER

- 第4パラメタ名「outdata1」

項目名(パラメタ名)	指定内容
名前	outdata1
種別	出力
引数型	CharPTR
データ型	SQL_VARCHAR

- 第5パラメタ名「outdata2」

項目名(パラメタ名)	指定内容
名前	outdata2
種別	出力
引数型	CharPTR
データ型	SQL_VARCHAR

#### データ例

上記のプラグインを実行したときの各パラメタのデータ例です。各パラメタのデータは、以下のようになります。

パラメタ	定義の「種別」に指定した値	データ例
in_data	入力	20100110 (西暦、月、日のデータ (YYYYMMDD形式))

パラメタ	定義の「種別」に指定した値	データ例
in_string	入力	%4s%2s%2s パラメタ「in_data」のフォーマットにあわせて、書式化文字列を固定文字列として、データ変換定義から指定します。変換ごとに個別で指定することが可能です。
result	結果	2010 この例では、西暦のデータが返却されます。
out_data1	出力	01 アプリケーションインタフェースとしての出力という位置付けである引数です。返却される出力データ(結果)ではありません。
out_data2	出力	10 アプリケーションインタフェースとしての出力という位置付けである引数です。返却される出力データ(結果)ではありません。

## H.2.12 プラグイン定義とデータ変換定義の演算式の関係

データ変換定義の演算式には、プラグイン定義に指定したパラメタのデータ項目を指定します。

### ポイント

#### 演算式へのデータ項目の指定

プラグイン定義に指定した各パラメタの指定順序と、データ変換定義の演算式に指定する指定順序は、必ずあわせてください。

### 注意

#### 演算式へのデータ項目の指定が不要なパラメタについて

以下の条件に該当するパラメタに関してのみ、演算式へのデータ項目の指定は不要です。

- プラグイン定義の指定で、種別に「結果」を指定したパラメタ  
変換プラグイン機能が実行結果値を変換ファンクションに返却するためです。
- プラグイン定義の指定で、種別に「デフォルト」を指定したパラメタ  
変換プラグイン機能で定義値をアプリケーションに渡すためです。

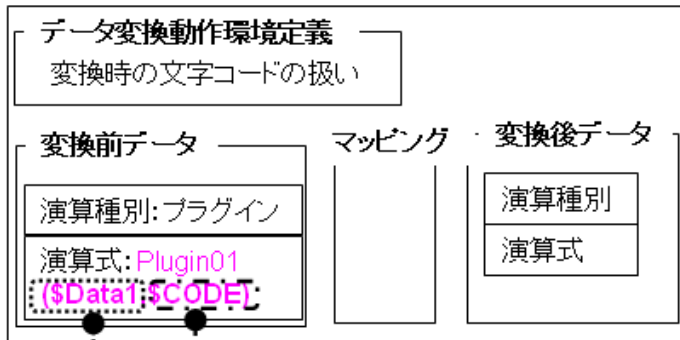
### 例

“H.2.11 アプリケーションとプラグイン定義の例”で説明した例2のアプリケーションで、プラグイン定義とデータ変換定義の関係を示します。

この例では、デザインシートのプラグイン定義シートとデータ変換定義シートの関係を示しています。プラグイン定義の第3パラメタ、第4パラメタのデータ項目を、それぞれデータ変換定義の演算式に指定します。

プラグイン定義の指定で第1パラメタは「結果」、第2パラメタは「デフォルト」であるため、演算式へのデータ項目の指定は不要です。

## データ変換定義シート



この例では、  
データ変換定義シートへの  
演算式への定義は不要

パラメタ3に関して指定する。  
データ項目をデータ変換定義シートの演算式に。  
パラメタ情報はプラグイン定義シートに。

パラメタ4に関して指定する。  
データ項目をデータ変換定義シートの演算式に。  
パラメタ情報はプラグイン定義シートに。

## プラグイン定義シート

プラグインID: Plugin01
プラグインパスD:\...
プラグイン種別: DLL
復帰値の属性: Int
関数名: setstring_func
フォーマットID: format1
パラメタ情報1: out_data ...
<b>パラメタ情報1</b>
名前: out_data
種別: 結果
引数型: CharPTR
データ型: SQL_VARCHAR
<b>パラメタ情報2</b>
名前: out_format
種別: デフォルト
引数型: CharPTR
デフォルト設定値: No(%d) Data(%s)
データ型: SQL_VARCHAR
<b>パラメタ情報3</b>
名前: code_no
種別: 入力
引数型: Long
データ型: SQL_INTEGER
<b>パラメタ情報4</b>
名前: in_data
種別: 入力
引数型: CharPTR
データ型: SQL_VARCHAR

# 付録I プラグイン機能を利用した帳票出力

プラグイン機能を利用して行う帳票出力について説明します。

## ポイント

### プラグイン機能を利用した帳票出力

- Information Integratorサーバと同じ場所に配置したInterstage List Creatorにのみ出力することができます。その他は、Information Integratorのデータ配付機能で直接、帳票出力する場合とすべて同一の動作となります。
- 帳票出力は、出力するデータを一度データボックスに配付し、配付したデータをプラグイン機能でInterstage List Creatorへ帳票データを渡す手順となります。

## I.1 データボックスにデータを配付する

III Studioを利用して行う場合は、“[6.7 データ配付処理の設計](#)”を参照してください。

データボックスにデータを配付する方法は、デザインシートを利用して行う場合は、“[7.5 データ配付処理の設計](#)”を参照してください。

## I.2 帳票データを出力するプラグインの設計

Interstage List Creatorへ帳票データを渡すためのプラグインの設計を行います。

### III Studioを利用して行う場合

- 処理プラグイン機能の設計手順  
詳細は、“[6.10.1 処理プラグイン機能の設計手順](#)”を参照してください。
- 帳票出力時のプラグイン定義作成方法  
プラグイン定義の作成時に、出力する帳票に応じたコマンドを指定します。ここでは、コマンドの内容について説明します。プラグイン定義へのコマンドの入力方法は、“[III Studio 操作ヘルプ](#)”を参照してください。

### デザインシートを利用して行う場合

- 処理プラグイン機能の設計手順  
詳細は、“[7.8.1 処理プラグイン機能の設計手順](#)”を参照してください。
- 帳票出力時のプラグイン定義シート作成方法  
プラグイン定義シートの作成時に、出力する帳票に応じたコマンドを指定します。コマンドの内容について説明します。プラグイン定義シートへのコマンドの入力方法は、“[デザインシート操作ヘルプ](#)”を参照してください。

### I.2.1 ifilcprints(シングルフォーム出力コマンド)

#### 説明

Interstage List Creatorでシングルフォーム出力するコマンドです。

#### 注意

Interstage List Creatorへ渡した帳票データは、画面对話が必要なプリンタには出力できません。画面对話が不要なプリンタに出力してください。

## コマンドの格納先

- Windowsの場合

「ifilcprints.cmd」は、以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥enabler¥bin
```

- UNIXの場合

「ifilcprints.sh」は、以下のディレクトリにあります。

```
/opt/FJISVifisv/enabler/bin/
```

## コマンドの形式

```
ifilcprints 帳票名  
            データボックス名  
            [-LCOPT 拡張オプション]
```

## オペランドの説明

### 帳票名

帳票名を指定します。本指定は、必須です。帳票名は、英数字のみ使用可能です。

### データボックス名

データ配付定義で指定したデータボックス名を指定します。本指定は、必須です。

### 拡張オプション

本指定は、任意です。

本指定を行う場合は、先頭に必ず「-LCOPT」を指定してください。「-LCOPT」がない場合、本指定は無効となります。

指定可能なオプションは、Interstage List Creatorの「prprint(帳票出力)」コマンドに指定可能なオプションです。指定可能な拡張オプションの詳細は、Interstage List Creatorのマニュアルを参照してください。


## 注意

- Interstage List Creatorの「prprint(帳票出力)」コマンドに指定可能なオプションのうち、以下のオプションは指定できません。
  - 「帳票名」
  - 「-f」
  - 「-grpdelimit」
- 拡張オプションに指定可能なオプションの中には、帳票様式定義でも指定可能なオプションがあります。これらのオプションは、拡張オプションまたは帳票様式定義のどちらかに指定があれば有効となります。拡張オプションおよび帳票様式定義の両方に指定された場合は、拡張オプションの指定が有効となります。
- Windowsの場合、プリンタへの出力はInformation Integratorサービスのログオンアカウントの権限で行われます。プリンタを利用できるユーザが制限されている場合は、Information Integratorサービスのログオンアカウントに設定されているユーザをプリンタの出力設定に追加(権限付与)してください。
- ディレクトリ、およびファイルについて、利用可能な名前の長さは以下のとおりです。

項目名	最大値
出力ファイル名	255バイト(絶対パス)
	63バイト(ファイル名のみ)
出力ディレクトリ名	191バイト(絶対パス)

## コマンドの復帰値

コマンドの復帰値、意味、対処方法は以下のとおりです。

復帰値	意味	対処方法
0	帳票の出力が正常に終了しました。	なし
<ul style="list-style-type: none"> <li>• Windows 1101</li> <li>• UNIX 11</li> </ul>	Interstage List Creatorでエラーが発生しました。	<p>Windowsの場合はイベントログ、UNIXの場合はシステムログを確認し、対処を行ってください。</p> <p>対処方法の詳細については、Interstage List Creatorのマニュアルを参照してください。</p> <p> <b>ポイント</b></p> <p>.....</p> <p>帳票データの出力時にエラーが発生した場合は、イベントログ (Windowsの場合) やシステムログ (UNIXの場合) にInterstage List Creatorのメッセージが出力されることがあります。Interstage List Creatorのメッセージの対処方法については、Interstage List Creatorのマニュアルを参照してください。</p> <p>.....</p> <p>なお、本エラーは、帳票の出力先のプリンタやディレクトリを利用する権限がない場合にも発生します。各種権限に関する設定については、“9.5 帳票出力利用時のセキュリティ設定情報”を参照してください。</p>
<ul style="list-style-type: none"> <li>• Windows 1102</li> <li>• UNIX 12</li> </ul>	データボックスからのデータ読み込みでエラーが発生しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• 「データボックス名」が正しく指定されているか確認してください。</li> <li>• 作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>• 作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 1201</li> <li>• UNIX 21</li> </ul>	必須オペランドが指定されていません。	オペランドに「帳票名」、「データボックス名」を正しく指定してください。
<ul style="list-style-type: none"> <li>• Windows 1301</li> <li>• UNIX 31</li> </ul>	作業ファイルの作成に失敗しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• 「データボックス名」が正しく指定されているか確認してください。</li> <li>• 作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>• 作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>

## I.2.2 ifilcprintm(マルチフォーム出力コマンド)

### 説明

CSVファイル形式で配付データボックスに格納されたデータを、Interstage List Creatorでマルチフォーム出力するコマンドです。

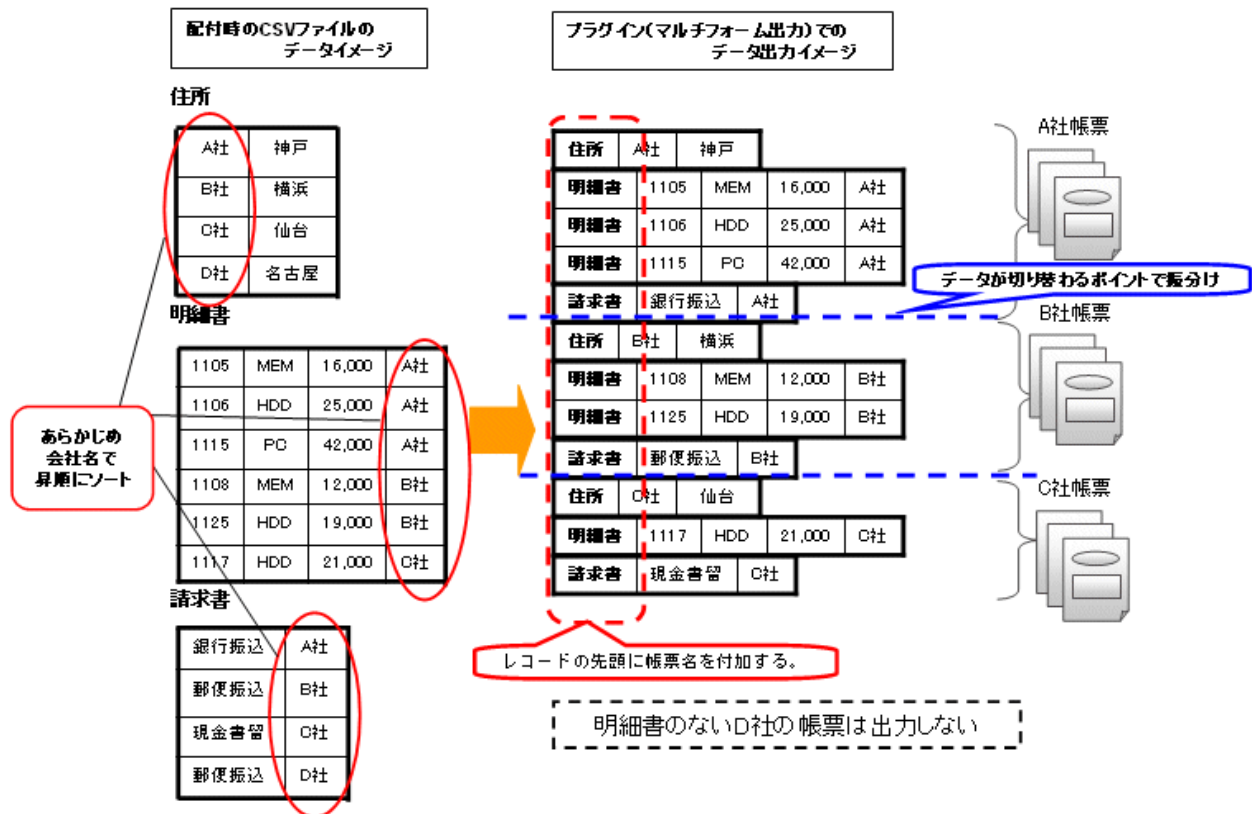


## 注意

Interstage List Creatorへ渡した帳票データは、画面对話が必要なプリンタには出力できません。画面对話が不要なプリンタに出力してください。

マルチフォームで3種類の帳票(住所、明細書、請求書)を会社単位(A社、B社、C社、D社)に出力するときのデータ作成イメージは以下のとおりです。

図I.1 マルチフォーム出力時のデータ作成イメージ



本コマンドで、以下の処理を行います。

1. 帳票振り分け  
レコード内のデータが切り替わるポイント(A社→B社→C社)でデータを振り分ける
2. 帳票名の付加  
データの各レコード(行)の先頭カラムに帳票名(帳票定義体名)を設定する。
3. データ結合  
宛先(A社、B社、C社、D社)毎に分割した住所、明細書、請求書を結合する。
4. 振り分けデータのない帳票  
明細書のないD社の帳票は出力しないように住所、請求書のデータを削除する。
5. 帳票出力  
帳票を出力する。

## ポイント

本コマンドを利用する前に

すべてのデータは、同じ方式によって振り分けるキーとなる項目でソートを行ってください。

## コマンドの格納先

- Windowsの場合

「ifilcprintm.cmd」は、以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥enabler¥bin
```

- UNIXの場合

「ifilcprintm.sh」は、以下のディレクトリにあります。

```
/opt/FJSVifisv/enabler/bin/
```

## コマンドの形式

```
ifilcprintm 入力データ形式
            データ区切り文字
            全帳票数
            帳票名n 振分けキー項目番号n 振分けデータがない時の動作方法n データボックス名n
            [-LCOPT 拡張オプション]
```

## オペランドの説明

### 入力データ形式

Interstage List Creatorへの入力データ形式を指定します。Interstage List Creatorの帳票様式定義に従って、以下のいずれかを指定します。本指定は、必須です。

帳票様式定義の指定値	本コマンドの指定値
CSV形式	CSV
可変長	VAR

### データ区切り文字

データの区切り文字を指定します。本指定は、必須です。

カンマの場合:COM

タブの場合:TAB

スペースの場合:SPA



必ずデータの区切り文字にあわせて指定してください。異なる区切り文字を指定すると、正しく動作しません。

### 全帳票数

結合する帳票の数(2以上6以下)を指定します。本指定は、必須です。

### 帳票名n 振分けキー項目番号n 振分けデータがない時の動作n データボックス名n

帳票名、振分けキー項目番号、振分けデータがない時の時の動作およびデータボックス名を、全帳票数で指定した数だけ指定します。マルチフォームに出力する順に指定します。本指定は、必須です。

### 帳票名

帳票名を指定します。帳票名は、英数字のみ使用可能です。

## 振分けキー項目番号

全帳票数で指定した帳票それぞれに対して、振分けを行うキーとなる項目の番号を指定します。左から順に1、2・・・と数えます(最大は1024)。すべてのデータは振り分けのキーとなる項目で同じ方式でソートが行われている必要があります。

## 振分けデータがない時の動作

振分けデータがない時の動作方法をM(マスタ)または、D(明細)より選択して指定します。

### M(マスタ)

振分けキー項目には他のM(マスタ)またはD(明細)の振分けキー項目に存在するすべての値が存在していなければなりません。

### D(明細)

Dを指定したすべての帳票で同じ振分け値に対するデータが0件だった場合、その振分け値に対するデータについてはM(マスタ)を指定した帳票も含めてすべての帳票を出力しません。

## データボックス名

データ配付定義で指定したデータボックス名を指定します。

## 拡張オプション

本指定は、任意です。

本指定を行う場合は、先頭に必ず「-LCOPT」を指定してください。「-LCOPT」がない場合、本指定は無効となります。

指定可能なオプションは、Interstage List Creatorの「prprint(帳票出力)」コマンドに指定可能なオプションです。指定可能な拡張オプションの詳細は、Interstage List Creatorのマニュアルを参照してください。




- Interstage List Creatorの「prprint(帳票出力)」コマンドに指定可能なオプションのうち、以下のオプションは指定できません。
  - 「帳票名」
  - 「-f」
  - 「-grpdelimit」
- 拡張オプションに指定可能なオプションの中には、帳票様式定義でも指定可能なオプションがあります。これらのオプションは、拡張オプションまたは帳票様式定義のどちらかに指定があれば有効となります。拡張オプションおよび帳票様式定義の両方に指定された場合は、拡張オプションの指定が有効となります。
- Windowsの場合、プリンタへの出力はInformation Integratorサービスのログオンアカウントの権限で行われます。プリンタを利用できるユーザが制限されている場合は、Information Integratorサービスのログオンアカウントに設定されているユーザをプリンタの出力設定に追加(権限付与)してください。
- ディレクトリ、およびファイルについて、利用可能な名前の長さは以下のとおりです。

項目名	最大値
出力ファイル名	255バイト(絶対パス)
	63バイト(ファイル名のみ)
出力ディレクトリ名	191バイト(絶対パス)

## コマンドの復帰値

コマンドの復帰値、意味、対処方法は以下のとおりです。コマンドの復帰値は、プラグイン実行時の詳細エラーコードを参照してください。

復帰値	意味	対処方法
0	帳票の出力が正常に終了しました。	なし
<ul style="list-style-type: none"> <li>• Windows 2101</li> <li>• UNIX 101</li> </ul>	Interstage List Creatorでエラーが発生しました。	<p>Windowsの場合はイベントログ、UNIXの場合はシステムログを確認し、対処を行ってください。</p> <p>対処方法の詳細については、Interstage List Creatorのマニュアルを参照してください。</p> <p> <b>ポイント</b></p> <p>.....</p> <p>帳票データの出力時にエラーが発生した場合は、イベントログ (Windowsの場合) やシステムログ (UNIXの場合) にInterstage List Creatorのメッセージが出力されることがあります。Interstage List Creatorのメッセージの対処方法については、Interstage List Creatorのマニュアルを参照してください。</p> <p>.....</p> <p>なお、本エラーは、帳票の出力先のプリンタやディレクトリを利用する権限がない場合にも発生します。各種権限に関する設定については、“<a href="#">9.5 帳票出力利用時のセキュリティ設定情報</a>”を参照してください。</p>
<ul style="list-style-type: none"> <li>• Windows 2102</li> <li>• UNIX 102</li> </ul>	データボックスからのデータ読み込みでエラーが発生しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• 正しい「データボックス名」が指定されているか確認してください。</li> <li>• 作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>• 作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2103</li> <li>• UNIX 103</li> </ul>	作業ファイルの作成に失敗しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• 作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>• 作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2111</li> <li>• UNIX 111</li> </ul>	オペランド「入力データ形式」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「入力データ形式」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2112</li> <li>• UNIX 112</li> </ul>	オペランド「データ区切り文字」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「データ区切り文字」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2113</li> <li>• UNIX 113</li> </ul>	オペランド「全帳票数」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「全帳票数」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>

復帰値	意味	対処方法
<ul style="list-style-type: none"> <li>• Windows 2114</li> <li>• UNIX 114</li> </ul>	オペランド「帳票名」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「帳票名」が正しく指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2115</li> <li>• UNIX 115</li> </ul>	オペランド「振り分けキー項目番号」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「振り分けキー項目番号」が正しく指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2116</li> <li>• UNIX 116</li> </ul>	オペランド「振り分けデータなし時動作」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「振り分けデータなし時動作」が正しく指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2117</li> <li>• UNIX 117</li> </ul>	オペランド「データボックス名」が指定されていません。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「データボックス名」が正しく指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2201</li> <li>• UNIX 131</li> </ul>	必須オペランドが指定されていません。	必須オペランドが正しく指定されているか確認してください。
<ul style="list-style-type: none"> <li>• Windows 2202</li> <li>• UNIX 132</li> </ul>	作業ファイル名の長さが最大値を越えました。	環境変数IFILC_TMPDIRで指定したディレクトリ名を短くしてください。
<ul style="list-style-type: none"> <li>• Windows 2203</li> <li>• UNIX</li> </ul>	作業ディレクトリ名の長さが最大値を越えました。	環境変数IFILC_TMPDIRで指定したディレクトリ名を短くしてください。

復帰値	意味	対処方法
133		
<ul style="list-style-type: none"> <li>• Windows 2204</li> <li>• UNIX 134</li> </ul>	オペランド「入力データ形式」の指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「入力データ形式」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2205</li> <li>• UNIX 135</li> </ul>	オペランド「データ区切り文字」の指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「データ区切り文字」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2206</li> <li>• UNIX 136</li> </ul>	オペランド「全帳票数」の指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「全帳票数」が正しく指定されているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2207</li> <li>• UNIX 137</li> </ul>	オペランド「全帳票数」が最大値を超えています。	オペランド「全帳票数」の値を小さくしてください。
<ul style="list-style-type: none"> <li>• Windows 2208</li> <li>• UNIX 138</li> </ul>	オペランド「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2209</li> <li>• UNIX 139</li> </ul>	オペランド「帳票名」の長さが最大値を超えました。	オペランド「帳票名」が正しく指定されているか確認してください。
<ul style="list-style-type: none"> <li>• Windows 2210</li> <li>• UNIX 140</li> </ul>	オペランド「振り分けキー項目番号」の指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>• オペランド「振り分けキー項目番号」が正しく指定されているか確認してください。</li> <li>• 「帳票名」、「振り分けキー項目番号」、「振り分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>• すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>

復帰値	意味	対処方法
<ul style="list-style-type: none"> <li>Windows 2211</li> <li>UNIX 141</li> </ul>	オペランド「振分けキー項目番号」が最大値を超えました。	オペランド「振分けキー項目番号」を小さくしてください。
<ul style="list-style-type: none"> <li>Windows 2212</li> <li>UNIX 142</li> </ul>	オペランド「振分けキー項目番号」の指定に誤りがあります。	オペランド「振分けキー項目番号」を数値で指定してください。
<ul style="list-style-type: none"> <li>Windows 2213</li> <li>UNIX 143</li> </ul>	オペランド「振分けデータなし時動作」の指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>オペランド「振分けデータなし時動作」が正しく指定されているか確認してください。</li> <li>「帳票名」、「振分けキー項目番号」、「振分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>Windows 2214</li> <li>UNIX 144</li> </ul>	オペランド「振分けデータなし時動作」の指定に誤りがあります。	オペランド「振分けデータなし時動作」が正しく指定されているか確認してください。
<ul style="list-style-type: none"> <li>Windows 2215</li> <li>UNIX 145</li> </ul>	オペランドの指定に誤りがあります。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>「帳票名」、「振分けキー項目番号」、「振分けデータなし時動作」、「データボックス名」の組の数が「全帳票数」と一致しているか確認してください。</li> <li>すべてのオペランドについて正しい順番で指定されているか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>Windows 2216</li> <li>UNIX 146</li> </ul>	作業ファイル名の長さが最大値を超えました。	環境変数IFILC_TMPDIRで指定したディレクトリ名を短くしてください。
<ul style="list-style-type: none"> <li>Windows 2301</li> <li>UNIX 151</li> </ul>	作業ファイルの作成に失敗しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>Windows 2302</li> <li>UNIX 152</li> </ul>	作業ファイルの作成に失敗しました。	<p>以下の対処を行ってください。</p> <ul style="list-style-type: none"> <li>作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>

復帰値	意味	対処方法
<ul style="list-style-type: none"> <li>• Windows 2303</li> <li>• UNIX 153</li> </ul>	「振分けデータなし時動作」に「M」を指定した帳票の入力データ(データボックス)に0バイトのものがあります。	「振分けデータなし時動作」に「M」を指定した帳票の入力データ(データボックス)は0バイトにならないようにしてください。
<ul style="list-style-type: none"> <li>• Windows 2304</li> <li>• UNIX 154</li> </ul>	「振分けデータなし時動作」に「M」が指定された帳票がありません。	「振分けデータなし時動作」には1つ以上の「M」を指定してください。
<ul style="list-style-type: none"> <li>• Windows 2305</li> <li>• UNIX 155</li> </ul>	「振分けデータなし時動作」に「M」を指定した帳票の入力データ(データボックス)に振分け値に異常のあるデータがあります。	データボックスに正しいデータを格納してください。
<ul style="list-style-type: none"> <li>• Windows 2306</li> <li>• UNIX 156</li> </ul>	出力データの1行の長さが最大値を超えました。	データの1行の長さを短くしてください。
<ul style="list-style-type: none"> <li>• Windows 2307</li> <li>• UNIX 157</li> </ul>	作業ファイルへの出力中にエラーが発生しました。	以下の対処を行ってください。 <ul style="list-style-type: none"> <li>• 作業ファイルの作成先ディレクトリに書き込み権限があるか確認してください。</li> <li>• 作業ファイルの作成先のディスクに十分な空き容量があるか確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2308</li> <li>• UNIX 158</li> </ul>	「振り分けキー項目番号」に0と0以外が混在しています。	「振り分けキー項目番号」はすべて0、または、すべて0以外で統一してください。
<ul style="list-style-type: none"> <li>• Windows 2309</li> <li>• UNIX 159</li> </ul>	「振り分けキー項目番号」に0以外と0が混在しています。	「振り分けキー項目番号」はすべて0、または、すべて0以外で統一してください。
<ul style="list-style-type: none"> <li>• Windows 2310</li> <li>• UNIX 160</li> </ul>	振り分けできないデータ行があります。	<ul style="list-style-type: none"> <li>• 「振り分けデータなし時動作」に「D」を指定したデータの振り分け値が、「M」を指定したデータの振り分け値の中に存在するか確認してください。</li> <li>• すべてのデータボックスのデータが、振り分けキー項目によって同じ方式でソートされているかを確認してください。</li> </ul>
<ul style="list-style-type: none"> <li>• Windows 2401</li> <li>• UNIX 171</li> </ul>	入力データの1行の長さが最大値を超えました。	入力データの1行の長さを短くしてください。



復帰値	意味	対処方法
<ul style="list-style-type: none"> <li>• Windows 2402</li> <li>• UNIX 172</li> </ul>	入力データに「振り分けキー項目番号」で指定した項目が見つかりません。	以下の対処を行ってください。 「振り分けキー項目番号」と「データボックス」の組み合わせが合っているか確認してください。
<ul style="list-style-type: none"> <li>• Windows 2403</li> <li>• UNIX 173</li> </ul>	入力データに「振り分けキー項目番号」で指定した項目が見つかりません。	以下の対処を行ってください。 「振り分けキー項目番号」と「データボックス」の組み合わせが合っているか確認してください。
<ul style="list-style-type: none"> <li>• Windows 2404</li> <li>• UNIX 174</li> </ul>	入力データで項目の終わりの囲み文字の後ろに区切り文字がありません。	入力データの各項目では囲み文字の直後に区切り文字がくるようにしてください。
<ul style="list-style-type: none"> <li>• Windows 2406</li> <li>• UNIX 176</li> </ul>	入力データで項目の途中で改行が検出されました。	入力データの各項目の前後が囲み文字になっているか確認してください。
<ul style="list-style-type: none"> <li>• Windows 2501</li> <li>• UNIX 181</li> </ul>	作業用のメモリ確保に失敗しました。	メモリが枯渇していないか確認を行ってください。

# 付録J サービスバス連携を行う場合のService Integratorの設定

サービスバス連携入力または、サービスバス連携出力を行う場合のService Integratorの設定について説明します。

なお、ここでは、Service Integrator 9.3の操作を基に、ポイントだけを説明します。操作方法などの詳細は、Interstage Service Integratorのマニュアルを参照してください。Service Integrator 9.4以降の操作については、サービスエンドポイントの接続方式に「III連携」を選択できるため、簡易に設定できます。操作方法などの詳細は、Interstage Service Integratorのマニュアルを参照してください。

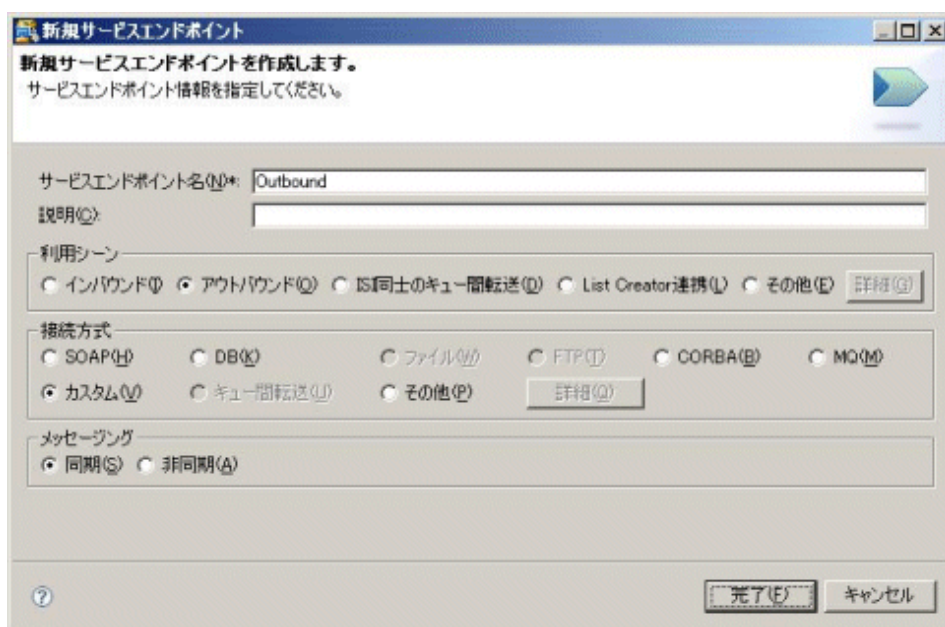
## J.1 サービスバス連携入力を行う場合

サービスバス連携入力を行う場合のService Integratorの設定について説明します。

### 設定個所と設定内容

- ・ [新規サービスエンドポイント]画面での設定

ISI Studioの[新規サービスエンドポイント]画面で、サービスエンドポイントを作成します。



以下のように設定してください。

項目	設定値	備考
利用シーン	「アウトバウンド」を選択	
接続方式	「カスタム」を選択	
メッセージング	「同期」を選択	必ず「同期」を設定してください。「非同期」を設定すると、接続できません。

- ・ [プロパティ]画面での設定

ISI Studioの[新規サービスエンドポイント]に関する情報を示す[プロパティ]画面で、サービスエンドポイントに関する各種情報を入力します。

以下のように設定してください。

ー リソースアダプタ

▼ リソースアダプタ

JNDI名\*: java:comp/env/eis/esiiiiadapter

ConnectionSpec実装クラス名\*:

ConnectionSpecのパラメータ

追加 削除

メソッド名	型	値

InteractionSpec実装クラス名\*: com.fujitsu.ifl.adp.isi.outbound.cci.IFLInteractionSpec

InteractionSpecのパラメータ

追加 削除

メソッド名	型	値
setProcessId	String	Process
setDataBox	String	DUMMY01

Interactionの実行メソッド形式:  boolean戻り型  Record戻り型

項目	設定値
JNDI名	java:comp/env/eis/esiiiiadapter
InteractionSpec実装クラス名	com.fujitsu.ifl.adp.isi.outbound.cci.IFLInteractionSpec
InteractionSpecのパラメータ	<ul style="list-style-type: none"> <li>メソッド名: setProcessId 型: String 値: プロセスIDを指定</li> <li>メソッド名: setDataBox 型: String 値: データボックス名を指定</li> </ul>
Interactionの実行メソッド形式	「boolean戻り型」を選択

ー インターフェイス調整

▼ インターフェイス調整

入力Recordの形式:  Indexed  Mapped

アダプタ呼び出し元の文字コード\*: UTF-8

入力Recordのパラメータ

追加 削除

Recordのキー名	共通メッセージからの値取得元	共通メッセージから取得する値の格納位置
com.fujitsu.esi.jcapayload	バイナリ型でペイロード	
com.fujitsu.esi.message.id	ヘッダ	com.fujitsu.esi.message.id

出力Recordの形式:  Indexed  Mapped

アダプタ呼び出し結果の文字コード\*: UTF-8

出力Recordのパラメータ

追加 削除

Recordのキー名	共通メッセージへの値設定先	共通メッセージへの値の格納位置
com.fujitsu.esi.message.id	ヘッダ	com.fujitsu.esi.message.id

項目	設定値
入力Recordの形式	「Mapped」を選択
アダプタ呼出し元の文字コード	「UTF-8」を選択
入力Recordのパラメタ	<ul style="list-style-type: none"> <li>• Recordのキー名 : com.fujitsu.esi.jca.payload <ul style="list-style-type: none"> <li>— 共通メッセージからの値取得元: バイナリ型ペイロード</li> </ul> </li> <li>• Recordのキー名 : com.fujitsu.esi.message.id <ul style="list-style-type: none"> <li>— 共通メッセージからの値取得元: ヘッダ</li> <li>— 共通メッセージから取得する値の格納位置 : com.fujitsu.esi.message.id</li> </ul> </li> </ul>
出力Recordの形式	「Mapped」を選択
アダプタ呼出し結果の文字コード	「UTF-8」を選択
出力Recordのパラメタ	<ul style="list-style-type: none"> <li>• Recordのキー名 : com.fujitsu.esi.message.id <ul style="list-style-type: none"> <li>— 共通メッセージへの値設定先: ヘッダ</li> <li>— 共通メッセージへの値の格納位置 : com.fujitsu.esi.message.id</li> </ul> </li> </ul>



### 注意

#### サービスエンドポイントを呼び出すシーケンスの名前について

シーケンス名は、36文字以下で指定してください。36文字を超える長さの名前を指定すると、データボックス格納時にデータの格納に失敗することがあります。

## J.2 サービスバス連携出力を行う場合

サービスバス連携出力を行う場合のService Integratorの設定について説明しています。

## 設定個所と設定内容

- ・ [新規サービスエンドポイント]画面での設定

ISI Studioの[新規サービスエンドポイント]画面で、サービスエンドポイントを作成します。

以下のように設定してください。

項目	設定値	備考
利用シーン	「インバウンド」を選択	
接続方式	「カスタム」を選択	
メッセージング	「同期」を選択	必ず「同期」を設定してください。「非同期」を設定すると、接続できません。

- ・ [プロパティ]画面での設定

ISI Studioの[新規サービスエンドポイント]に関する情報を示す[プロパティ]画面で、接続するシーケンス名を入力します。

以下のように設定してください。

### ー 一般

シーケンス名: シーケンス名は、36文字以下で指定してください。36文字を超える長さの名前を指定すると、データボックス格納時にデータの格納に失敗することがあります。

## 付録K プラグイン機能を利用したExcel連携

プラグイン機能を利用したExcel連携について説明します。

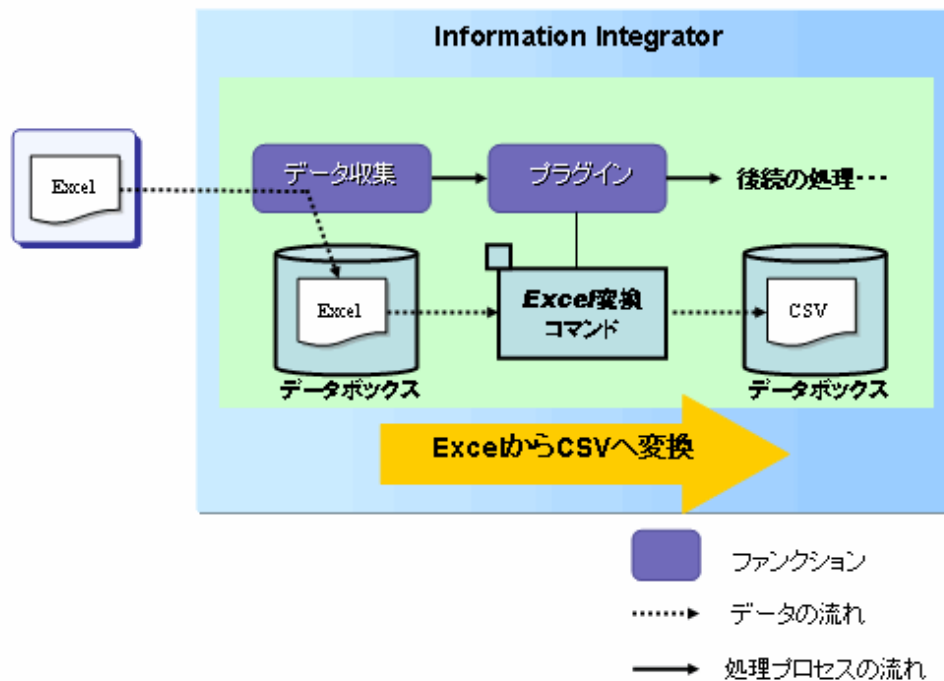
Excelファイルを集めたあと、指定したセル内のデータをCSVデータに変換することができます。また、CSVデータからExcelファイルに変換して配付することもできます。

これにより、他のデータ形式との連携が可能となります。

- Excel連携入力を行う場合

収集したExcelデータを後続のプラグインによってCSVデータに変換することで各種データ変換を可能にします。

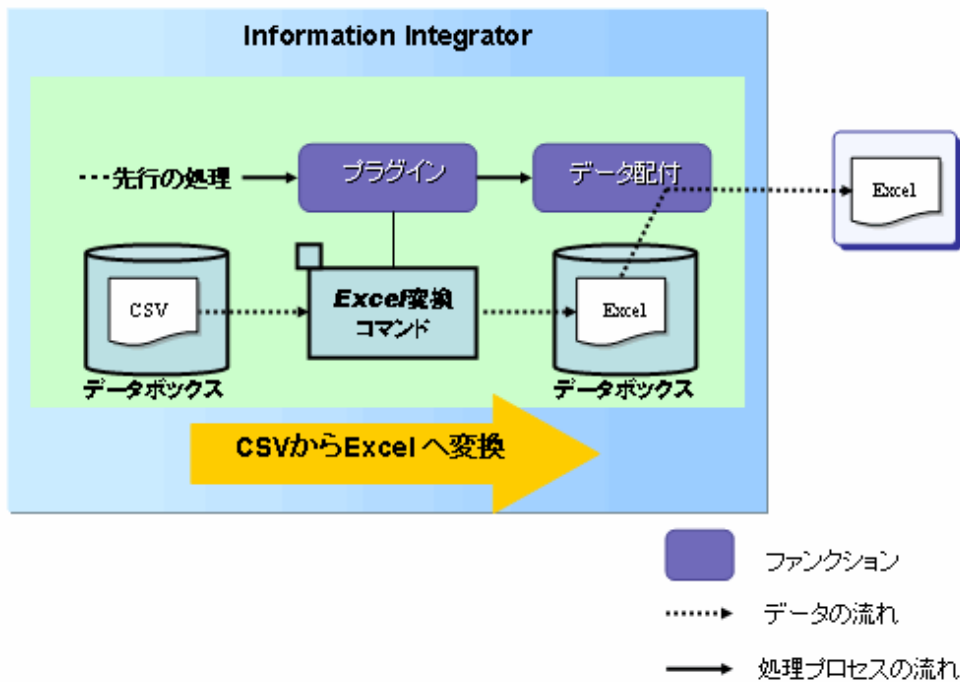
図K.1 Excel連携入力のイメージ



- Excel連携出力を行う場合

データ変換後のCSVデータを後続のプラグインによってExcelデータに出力し配付することを可能にします。

図K.2 Excel連携出力のイメージ



## P ポイント

データ収集およびデータ配付では、データ構造やフォーマットの設計は不要です。

データ収集およびデータ配付のデータボックス格納方法は「個別型」を指定してください。

## K.1 動作環境

Excel連携を使用する上での動作環境について説明します。

### K.1.1 Excelのバージョン

プラグインを実行する環境には、Excelがインストールされている必要があります。サポートするExcelのバージョンは以下の通りです。なお、複数のExcelがインストールされている場合、最後にインストールされたExcelが使用されます。

#### サポートするExcelのバージョン

- Excel 2003
- Excel 2007
- Excel 2010

#### サポートするExcelシート

操作可能なExcelファイルはインストール済みのExcelのバージョンに従います。また、各ExcelバージョンでサポートするExcelブックの形式を次に示します。

バージョン	Excel連携入力	Excel連携出力
Excel 2003	xls形式	xls形式

バージョン	Excel連携入力	Excel連携出力
Excel 2007	xls形式(※1)	xlsx形式(※2)
Excel 2010	xlsx形式	

※1 使用するExcelバージョンの形式に変換します。

※2 ifiexcelscsvconvコマンドの「-r」オペランドで参照するExcelファイルがxls形式の場合、使用するExcelバージョンの形式に変換して出力します。



## 注意

使用するExcelで扱うことのできないExcelバージョンの機能が含まれるデータの場合、Excel側でデータの変換が行われます。扱えない機能や振る舞いについては、Excelのヘルプを参照してください。

## プライマリ相互運用機能アセンブリ (PIA)

プラグインを実行する環境には、Excelのプライマリ相互運用機能アセンブリ (PIA)をインストールする必要があります。

また、プライマリ相互運用機能アセンブリ (PIA)をインストールするためには、「.NET Framework」をインストールする必要があります。これらの詳細を説明します。

- Excelプライマリ相互運用機能アセンブリのインストール

Excelプライマリ相互運用機能アセンブリは、Microsoft Officeのセットアップでインストールすることができます。

- 新規にExcelをインストールする場合

「.NET Framework」をインストールします。

Microsoft Officeのセットアップを起動します。

セットアップ時、「Microsoft Office Excel」のノードで「.Netプログラミングサポート」のオプションを「マイコンピュータから実行」または「マイコンピュータからすべて実行」に変更します。

- Excelがインストール済みの場合

「.NET Framework」をインストールします。

Microsoft Officeのセットアップを起動し、機能の追加/削除を実行します。

「Microsoft Office Excel」のノードで「.Netプログラミングサポート」のオプションを「マイコンピュータから実行」または「マイコンピュータからすべて実行」に変更します。



## 例

### Office 2003の場合

Office 2003の場合は以下の手順でインストールします。

- [Microsoft Office 2003 セットアップ] ダイアログ ボックスの [メンテナンス モード オプション] 画面で、[機能の追加/削除] をクリックし、[次へ] をクリックします。
- [アプリケーションごとにオプションを指定してインストール] チェック ボックスをオンにし、[次へ] をクリックします。  
アプリケーション固有のノードを展開します。  
たとえば、Microsoft Office Excel 2003 PIA を入手するには、[Microsoft Office Excel] ノードを展開し、[.NET プログラミング サポート] をクリックします。更新オプションを選択するために [NET プログラミング サポート] の隣のドロップダウン矢印をクリックし、[マイコンピュータから実行] をクリックします。[初めて実行するときにインストール] は選択しないでください。
- [更新] をクリックします。

- Excelプライマリ相互運用機能アセンブリの確認方法

次のいずれかのフォルダにおいて、アセンブリ名およびバージョンを確認してください。



Windowsインストールドライブ:¥WINDOWS¥assembly
Windowsインストールドライブ:¥WINNT¥assembly

Excelのバージョン	アセンブリ名	アセンブリのバージョン
Excel 2003	Microsoft.Office.Interop.Excel	11.0.0.0
Excel 2007	Microsoft.Office.Interop.Excel	12.0.0.0
Excel 2010	Microsoft.Office.Interop.Excel	14.0.0.0

- プライマリ相互運用機能のアセンブリに必要な「.NET Framework」のバージョン  
 プライマリ相互運用機能のアセンブリに必要な「.NET Framework」のバージョンは以下のとおりです。
  - Excel 2003:1.1以上
  - Excel 2007:1.1以上
  - Excel 2010:2.0以上

## K.2 利用可能なデータ

ここではExcel連携で利用可能なデータについて説明します。

ExcelからCSVへの変換では、Excelのセルに表示されている文字列を取り出します。入力されている値や計算式等は取り出せません。



### 例

次の場合、ExcelからCSVへの変換ではいずれも「10%」の文字列を取り出します。

- セルの書式設定で表示形式に「標準」を設定。セルに「10%」を入力し「10%」が表示されている場合。
- セルの書式設定で表示形式に「パーセンテージ」を設定。セルに「10」を入力し「10%」が表示されている場合。
- セルの書式設定で表示形式に「パーセンテージ」を設定。セルに「=1/10」を入力し「10%」が表示されている場合。

CSVからExcelへの変換では、CSVデータの文字列をExcelのセルに入力します。

CSVの形式を次に示します。

- 区切り文字  
カンマ
- 文字コード  
UTF16MS
- 囲み文字  
ダブルクォート



### 注意

- データ内に区切り文字や改行が入る場合は、囲み文字が必要です。
- データとしての囲み文字はエスケープ文字が必要です。また、連続する囲み文字2つで1つの囲み文字を表現します。  
 たとえば、CSV入力の場合は、「A"B」は「"A""B"」と表現します。



### ポイント

CSV形式のデータ変換の設計

CSV形式のデータ変換の設計をする場合、以下のように定義してください。

- フォーマット定義
  - 文字コード:UTF16MS
  - レコードタイプ:CSV
  - 区切り文字:カンマ
- データ構造定義
  - データ型:文字列属性として扱う場合は SQL\_CHAR、SQL\_VARCHAR  
数値属性として扱う場合は SQL\_INTEGER、SQL\_DECIMAL

---

## K.2.1 利用可能なデータにおける注意事項

---

Excel連携で利用可能なデータにおける注意事項について説明します。

### セルを結合している場合

- ExcelからCSVへの変換

ExcelからCSVへの変換では、結合前の状態で最初に出現するセルの位置のデータとして取り出します。

結合前の状態で2番目以降のセルの位置は、空のデータを取り出します。

結合されているセルの一部がセル範囲として指定されている場合、結合前の状態で最初に出現するセルの位置が指定の範囲内であればデータを取り出します。

範囲外の場合、空のデータを取り出します。



### 例

- 入力

	A	B	C	D	E
1	1	2	3	4	
2		22	33	44	
3					

- 出力

セル範囲:A1からD2

```
"1","2","3","4"  
,"22","33","44"
```

- 出力

セル範囲:A2からD2

```
,"22","33","44"
```

- 
- CSVからExcelへの変換

CSVからExcelへの変換では、参照または更新の場合に結合前の状態で最初に出現するセルの位置に対応するCSVデータを入力します。

結合前の状態で2番目以降のセルには、CSVデータは入りません。

結合されているセルの一部がセル範囲として指定されている場合、結合前の状態で最初に出現するセルの位置が指定の範囲内であれば、その位置に対応するCSVデータを入力します。範囲外の場合、CSVデータは入りません。

## 例

### ー 入力

```
"1","2","3","4"  
"11","22","33","44"
```

### ー 参照または更新対象のExcel

セル範囲:A1から出力

	A	B	C	D	E
1					
2					
3					

### ー 出力

	A	B	C	D	E
1	1	2	3	4	
2		22	33	44	
3					

## セル内で改行を含むデータを扱う場合

セル内で改行を含むデータを扱う場合、CSVに変換したときにそのまま改行が入ってしまうため、後続のファンクションでDB連携や変換などを行う場合には注意してください。

## マクロが設定されている場合

マクロの設定有無に関わらず、入力または取り出しを行います。

## 非表示のセルがある場合

フィルターなどで非表示のセルがある場合、指定されたセル範囲内であればセルに対して入力または取り出しを行います。

## データの先頭にシングルクォートがある場合

セル上にシングルクォートが表示されていなければ、シングルクォートは取り出しません。

## シートまたはセルの保護

CSVからExcelへの変換では、更新対象のシートまたはセルが保護されている場合はエラーになります。

ExcelからCSVへの変換では、入力のシートまたはセルが保護されていてもデータを取り出します。

## パスワード保護

Excel連携では、パスワードが設定されたExcelファイルは扱えません。

パスワード保護を解除したExcelファイルを使用してください。

## K.3 Excel連携用プラグインの設計

Excel連携に使用するプラグインの設計を行います。

### III Studioを利用して行う場合

- ・ 処理プラグイン機能の設計手順  
詳細は、“6.10.1 処理プラグイン機能の設計手順”を参照してください。
- ・ Excel連携時のプラグイン定義作成方法  
プラグイン定義の作成時に、データに応じたコマンドを指定します。ここでは、コマンドの内容について説明します。プラグイン定義へのコマンドの入力方法は、“III Studio 操作ヘルプ”を参照してください。

### デザインシートを利用して行う場合

- ・ 処理プラグイン機能の設計手順  
詳細は、“7.8.1 処理プラグイン機能の設計手順”を参照してください。
- ・ Excel連携時のプラグイン定義シート作成方法  
プラグイン定義シートの作成時に、データに応じたコマンドを指定します。ここでは、コマンドの内容について説明します。プラグイン定義シートへのコマンドの入力方法は、“デザインシート操作ヘルプ”を参照してください。

## K.3.1 ExcelデータをCSVデータに変換するプラグイン

ExcelデータをCSVデータに変換するプラグインについて説明します。

ExcelデータからCSVデータへ変換する場合、指定したセルの開始位置から終了位置までの範囲のデータを取得し、CSVデータに出力します。

なお、Excelの行と列はそれぞれCSVデータの行と列に対応します。

### ポイント

ExcelからCSVに変換すると、全てのデータに囲み文字が付与されます。

図K.3 入力Excelデータ

	A	B	C	D	E	F
1						
2		No.	商品名	価格	在庫数	
3		1	えんぴつ	90	15	
4		2	消しゴム	100	10	
5		3	ノート	150	5	
6		4	ボールペン	120	3	
7		5	三角定規	50	11	
8		6	コンパス	300	2	
9		7	クリップ	10	80	
10		8	ホッチキス	250	7	
11		9	付箋紙(青)	150	10	
12						

入力ExcelデータのB3-E7の範囲をCSVデータに出力した場合の出力CSVデータです。

図K.4 出力CSVデータ

```
"1","えんぴつ","90","15"  
"2","消しゴム","100","10"  
"3","ノート","150","5"  
"4","ボールペン","120","3"  
"5","三角定規","50","11"
```

## K.3.2 CSVデータをExcelデータに変換するプラグイン

CSVデータをExcel上の指定したセルを起点にして出力します。CSVデータの行と列はExcelの行と列にそれぞれに対応します。

図K.5 入力CSVデータ

```
"1","えんぴつ","90","15"  
"2","消しゴム","100","10"  
"3","ノート","150","5"  
"4","ボールペン","120","3"  
"5","三角定規","50","11"
```

入力CSVデータをExcel上のA1を起点に出力した場合の出力Excelデータです。

図K.6 出力Excelデータ

	A	B	C	D	E
1	1	えんぴつ	90	15	
2	2	消しゴム	100	10	
3	3	ノート	150	5	
4	4	ボールペン	120	3	
5	5	三角定規	50	11	
6					
7					
8					

### 既存データの更新

既存のExcelデータを使用して、元の書式や指定範囲外のデータを維持したままデータを書き換えることが可能です。

図K.7 入力CSVデータ

```
"11"  
"22"  
"33"  
"44"  
"55"  
"66"  
"77"  
"88"  
"99"
```

図K.8 既存Excelデータ

	A	B	C	D	E	F
1						
2		No.	商品名	価格	在庫数	
3		1	えんぴつ	90	0	
4		2	消しゴム	100	0	
5		3	ノート	150	0	
6		4	ボールペン	120	0	
7		5	三角定規	50	0	
8		6	コンパス	300	0	
9		7	クリップ	10	0	
10		8	ホッチキス	250	0	
11		9	付箋紙(青)	150	0	
12						

入力CSVデータを既存ExcelデータのE3を起点に更新した場合の出力Excelデータです。

図K.9 出力Excelデータ

	A	B	C	D	E	F
1						
2		No.	商品名	価格	在庫数	
3		1	えんぴつ	90	11	
4		2	消しゴム	100	22	
5		3	ノート	150	33	
6		4	ボールペン	120	44	
7		5	三角定規	50	55	
8		6	コンパス	300	66	
9		7	クリップ	10	77	
10		8	ホッチキス	250	88	
11		9	付箋紙(青)	150	99	
12						

## K.4 Excel連携で利用するコマンド

Excel連携で利用するコマンドについて説明します。

### K.4.1 ifiexclcsvcnv (Excel変換コマンド)

#### 説明

本コマンドはExcelデータからCSVデータに変換します。また、CSVデータからExcelデータに変換します。

#### コマンドの格納先

- Windowsの場合

以下のディレクトリにあります。

```
Information Integratorサーバインストールディレクトリ¥enabler¥bin
```

#### コマンドの形式

```
ifiexclcsvcnv [-v {2003 | 2007 | 2010}]
               [-m {excl2csv | csv2excl}]
               -i 入力ファイル名
```

-o 出力ファイル名  
-s 対象シート名  
[-a セル開始位置]  
[-b セル終了位置]  
[-d]  
[-u]  
[-r 参照ファイル名]  
[-n {null | skip}]

## オペランドの説明

### -v {2003 | 2007 | 2010}

Excel連携で利用するExcelのバージョンを以下の形式で指定します。省略すると、2010が採用されます。

2003: Excel 2003を利用します。

2007: Excel 2007を利用します。

2010: Excel 2010を利用します。

### -m {excl2csv | csv2excl}

変換の内容を以下の形式で指定します。省略すると、excl2csvが採用されます。

excl2csv: ExcelからCSVに変換します。

csv2excl: CSVからExcelに変換します。

### -i 入力ファイル名

入力ファイル名を指定します。以下のように指定します。

— 「-m」オペランドに「csv2excl」を指定した場合

255バイト以内の絶対パスで指定します。

— 「-m」オペランドに「excl2csv」を指定した場合

218バイト以内の絶対パスで指定します。

プラグインで前処理関クションのデータを入力とする場合は、「\$IFI\_InputFile」を指定してください。

本オペランドは、省略できません。

### -o 出力ファイル名

出力ファイル名を指定します。以下のように指定します。

— 「-m」オペランドに「csv2excl」を指定した場合

218バイト以内の絶対パスで指定します。

— 「-m」オペランドに「excl2csv」を指定した場合

255バイト以内の絶対パスで指定します。

プラグインの出力ファイルを後処理関クションのデータとする場合は、「\$IFI\_OutputFile」を指定してください。

本オペランドは、省略できません。

### -s 対象シート名

処理対象のExcelシート名を31バイト以内の半角または全角文字で指定します。本オペランドは、省略できません。

なお、以下の記号は使用できません。

∴、¥、?、[、]、/、\*

### -a セル開始位置

処理対象のExcelセル範囲の開始位置を指定します。省略すると、先頭(1行1列)が指定されたものとして動作します。

以下の形式で指定します。

一 A1形式またはR1C1形式 (R[行番号]C[列番号])

例えば、5行3列のExcelデータの場合にはR5C3と表現します。

行番号と列番号に指定できる数値はExcelのバージョンによって異なります。

- Excel 2003以前

行番号は1～65536の範囲で指定します。

列番号は1～256の範囲で指定します。

- Excel 2007以降

行番号は1～1048576の範囲で指定します。

列番号は1～16384の範囲で指定します。

**-b セル終了位置**

処理対象のExcelセル範囲の終了位置を指定します。以下の形式で指定します。

一 A1形式またはR1C1形式 (R[行番号]C[列番号])

行番号と列番号に指定できる数値はExcelのバージョンによって異なります。

- Excel 2003以前

行番号は開始位置～65536の範囲で指定します。

列番号は開始位置～256の範囲で指定します。

- Excel 2007以降

行番号は開始位置～1048576の範囲で指定します。

列番号は開始位置～16384の範囲で指定します。

 **注意**

「-m」オペランドに「excl2csv」を指定した場合、本オペランドは省略できません。

「-m」オペランドに「csv2excl」を指定した場合、本オペランドを省略すると、入力データが存在する分だけ出力します。指定時には不足分の入力をNULLデータとして扱い、指定範囲を超えたデータが入力された場合は異常とします。

**-d**

指定された出力ファイルが既に存在する場合、ファイルを新規に作成します。

「-u」オペランドと同時に指定することはできません。

 **注意**

本オペランドと「-u」オペランドが省略された場合に出力ファイルが存在すると、コマンドは異常終了します。

**-u**

指定された出力ファイルが既に存在する場合、シート、セルの内容を更新して上書き保存します。

出力ファイルが存在しない場合、ファイルを新規に作成します。また、シートが存在しない場合、シートを新規に作成します。

本オペランドは、「-m」オペランドに「csv2excl」を指定した場合にのみ指定できます。

なお、「-r」オペランドとは同時に使用できません。

また、「-d」オペランドと同時に指定することはできません。



## 注意

本オペランドと「-d」オペランドが省略された場合に出力ファイルが存在すると、コマンドは異常終了します。

### -r 参照ファイル名

Excel出力時に参照するExcelファイル名を218バイト以内の絶対パスで指定します。

指定されたExcelファイルに対し、「-s」オペランドで指定したシートのセルの内容を更新して、「-o」オペランドで指定したファイルに出力します。特定フォーマットのExcelファイルを元に、データを変化させて複数のExcel出力を行いたい場合などに使用します。「-m」オペランドに「csv2excel」を指定した場合にのみ指定可能です。

なお、「-u」オプションとは同時に使用できません。

### -n {null | skip}

CSVデータ中のNULLデータに対する扱いを指定します。省略すると、nullが採用されます。

本オペランドは、「-u」オペランドまたは「-r」オペランドを指定した場合に意味を持ちます。

「-m」オペランドに「csv2excel」を指定した場合にのみ指定可能です。

**null**:セルに空データを出力します。

**skip**:セルにデータを出力せず、既存のデータを維持します。

## コマンドの復帰値

0:正常

8:異常

## K.4.2 メッセージ

### DB15901

DB15901:Excel連携処理においてエラーが発生しました。詳細=detail

### メッセージ種別

エラー

### パラメタの意味

detail: 詳細エラーメッセージを出力します。

詳細エラーの意味と対処を、以下に示します。

## 注意

下記の詳細エラーメッセージ以外にExcelのメッセージが出力される場合があります。Excelのヘルプを参照して対処を実施してください。

- オプションの指定形式に誤りがあります。
  - 意味:オプションの指定形式に誤りがあります。
  - 対処:オプションの内容を確認してください。

- ファイルアクセスエラーが発生しました。(kind : name)
  - 意味:ファイルアクセスエラーが発生しました。
    - kind:エラー種別
      - NOT FOUND : ファイルが見つかりません。
      - WRITE : 書き込みができません。
      - READ : 読み込みができません。
    - name:ファイル名
  - 対処:指定したファイルにアクセス可能か確認してください。
- シートがありません。シート:sheet
  - 意味:指定されたExcelシートが見つかりません。
    - sheet:指定されたシート名
  - 対処:指定した対象シート名に誤りがないか確認してください。
- 出力ファイルが既に存在します。
  - 意味:指定された出力ファイルが既に存在します。
  - 対処:指定した出力ファイル名に誤りがないか確認してください。上書きする場合は「-d」または「-u」オプションを指定してください。
- 1項目のデータ長が制限を超えました。行:row 列:column
  - 意味:1項目のデータ長が最大長を超えました。
    - row:行番号
    - column:列番号
  - 対処:メッセージに出力された位置において、入力データに誤りがないか確認してください。
- 項目数が制限を超えました。行:row
  - 意味:入力データの項目数が出力可能な範囲を超えました。
    - row:行番号
  - 対処:メッセージに出力された位置において、入力データの項目数に誤りがないか確認してください。
- 行数が制限を超えました。
  - 意味:入力データの行数が出力可能な範囲を超えました。
  - 対処:入力データの行数に誤りがないか確認してください。
- EXCELが見つかりません。
  - 意味:Excelがインストールされていません。
  - 対処:正しいバージョンのExcelおよびプライマリ相互運用機能アセンブリがインストールされているか確認してください。
- 入力ファイル形式に誤りがあります。
  - 意味:入力データがExcel形式のデータではありません。
  - 対処:入力データに誤りがないか確認してください。
- CSVファイルの解析に失敗しました。行:row
  - 意味:入力データの解析で正しくデータを取得できませんでした。
    - row:行番号
  - 対処:メッセージに出力された位置において、入力データの形式に誤りがないか確認してください。

- 変更しようとしているセルまたはグラフは保護されているため、読み取り専用となっています。  
保護されているセルまたはグラフを変更するには、[校閲] タブをクリックし、[変更] グループの [シート保護の解除] をクリックして保護を解除します。パスワードの入力を要求されることもあります。
  - 意味:シートが保護されているため、データを書き込めません。
  - 対処:出力先シートを確認し、保護を解除してください。
- name にアクセスできません。
  - 意味:ファイルが使用中のため、アクセスできません。
    - name:ファイル名
  - 対処:処理対象のファイルが使用中でないか、確認してください。
- 別のプロセスで使用されているため、プロセスはファイル name にアクセスできません。
  - 意味:ファイルが使用中のため、アクセスできません。
    - name:ファイル名
  - 対処:処理対象のファイルが使用中でないか、確認してください。

## DB25901

DB25901:Excel連携処理が正常終了しました。

### メッセージ種別

情報

#### ポイント

##### DB25901の出力先について

DB25901のメッセージは、標準出力に出力されます。

## K.5 定義例

Excel連携のプラグイン定義の例を次に示します。

#### 参照

プラグイン定義の詳細は、“III Studioヘルプ”または“デザインシート操作ヘルプ”を参照してください。

### K.5.1 データ収集したExcelデータをCSVデータへ変換する場合

この場合の動作環境は以下のとおりです。

- Excelのバージョン  
Excel 2007
- Information Integratorサーバインストールディレクトリ  
C:\INTS\_II\SV
- シート名  
Sheet1

- 範囲

B3からE7の範囲をCSVデータへ変換します。

表K.1 プラグイン定義の定義内容

項目名(パラメタ名)	指定値
プラグインID	プラグインを識別する名前を指定します。
説明	定義に関する説明など、任意の文字を指定します。
プラグイン種別	「EXE」を指定します。
プラグインパス	C:\INTS_II\S\enabler\bin\ififexclscvncv.exe
入力パラメタ1-名前	-v
入力パラメタ2-名前	2007
入力パラメタ3-名前	-m
入力パラメタ4-名前	excl2csv
入力パラメタ5-名前	-i
入力パラメタ6-名前	\$IFI_InputFile
入力パラメタ7-名前	-o
入力パラメタ8-名前	\$IFI_OutputFile
入力パラメタ9-名前	-s
入力パラメタ10-名前	Sheet1
入力パラメタ11-名前	-a
入力パラメタ12-名前	B3
入力パラメタ13-名前	-b
入力パラメタ14-名前	E7

## K.5.2 データ変換したCSVデータをExcelデータへ変換する場合

この場合の動作環境は以下のとおりです。

- Excelのバージョン

Excel 2007

- Information Integratorサーバインストールディレクトリ

C:\INTS\_II\S\

- シート名

Sheet1

- 範囲

CSVデータをA1から出力します。

表K.2 プラグイン定義の定義内容

項目名(パラメタ名)	指定値
プラグインID	プラグインを識別する名前を指定します。
説明	定義に関する説明など、任意の文字を指定します。
プラグイン種別	「EXE」を指定します。
プラグインパス	C:\INTS_II\S\enabler\bin\ififexclscvncv.exe
入力パラメタ1-名前	-v

入力パラメタ2-名前	2007
入力パラメタ3-名前	-m
入力パラメタ4-名前	csv2excl
入力パラメタ5-名前	-i
入力パラメタ6-名前	\$IFI_InputFile
入力パラメタ7-名前	-o
入力パラメタ8-名前	\$IFI_OutputFile
入力パラメタ9-名前	-s
入力パラメタ10-名前	Sheet1
入力パラメタ11-名前	-a
入力パラメタ12-名前	A1

## K.6 運用時における注意事項

Excel連携の運用時の注意事項について説明します。

### K.6.1 プラグイン機能を利用したExcel連携のファンクション実行中の注意事項

プラグイン機能を利用したExcel連携のファンクションを実行中に、取り消し操作またはInformation IntegratorをFORCEモードで停止した場合、以下のプロセス(Windowsシステムのプロセス)が一時的に残存することがあります。

表K.3 プロセス一覧

ファンクション種別	プロセス名
プラグイン	ifiexclsvcnv2003.exe
	ifiexclsvcnv2007.exe
	ifiexclsvcnv2010.exe

残存したプロセスは時間が経過すると終了します。

なお、Information IntegratorをWARMモードで再起動する際に次のエラーメッセージが出力されることがありますが、以降のInformation Integratorの動作には影響はありません。また、COLDモードで再起動する場合はこのエラーメッセージは出力されません。

#### メッセージ

IF10220:他者が占有しているためにアクセス出来ません - 資源名(ファイル名)

#### 対処

- 起動前にタスクマネージャーでプロセスが存在していないかを確認してください。
- プロセスの終了を待たずにInformation IntegratorをWARMモードで再起動する場合は、タスクマネージャーよりプロセスを停止するかまたはシステムの再起動後に起動してください。

なお、COLDモードで再起動する場合は、上記対処は不要です。



#### 注意

FORCEモードでの停止前に実行中のExcel連携のファンクション状態は、Information IntegratorのWARMモードでの起動後も実行中(EXEC)と表示される場合があります。