XRF V3.8
ユーザーズマニュアル

株式会社ユニリタ

■本書の内容

本書は、XRFの概要と運用構成、主な機能について説明しています。

■本書の表記

本マニュアルではOSの基本的な操作についての記述は省略してあります。

[　]はメニュー名を表します。

[ファイル]-[開く]はメニューからプルダウンメニューのコマンドを選択することを表します。

<　>は、ダイアログ中のボタンを表します。

『　』は、参照先を表します。

本書中の斜体の文字には、値が設定されることを意味します。

例　value="*ファイル名*"

本書では、項目の選択肢を以下のように表しています。

{*A*｜*B*｜*C*}：{} 中の｜で区切られた*A*、*B*、*C*のうち、どれか1つを選択します。

A [,*B* [,*C*]]：[] で囲まれた項目は、省略することができます。たとえば、左記の場合の選択肢は、次の3つです。

・*A*を選択する。

・*A*、*B*を選択する。

・*A*、*B*および*C*を選択する。

マークについて

本書で使用するマークについて説明します。

このマークは、注意事項を表します。

このマークは、役に立つ情報やヒントを表します。

このマークは、制限事項を表します。

■商標について

XRF、DURL、ROF、FormHelperは、株式会社ユニリタの商標または登録商標です。

Microsoft、Windows、SQL Server、Excelは、米国Microsoft Corporationの米国およびその他の国における登録商標です。

Oracle、Oracle8i、Oracle9i、Java、JREは米国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。

DB2は、米国IBM Corporationの米国およびその他の国における登録商標です。

Acrobat Readerは、米国Adobe Systems Incorporatedの米国およびその他の国における登録商標です。

その他、記載されている社名および製品名は、各社の商標または登録商標です。

目 次

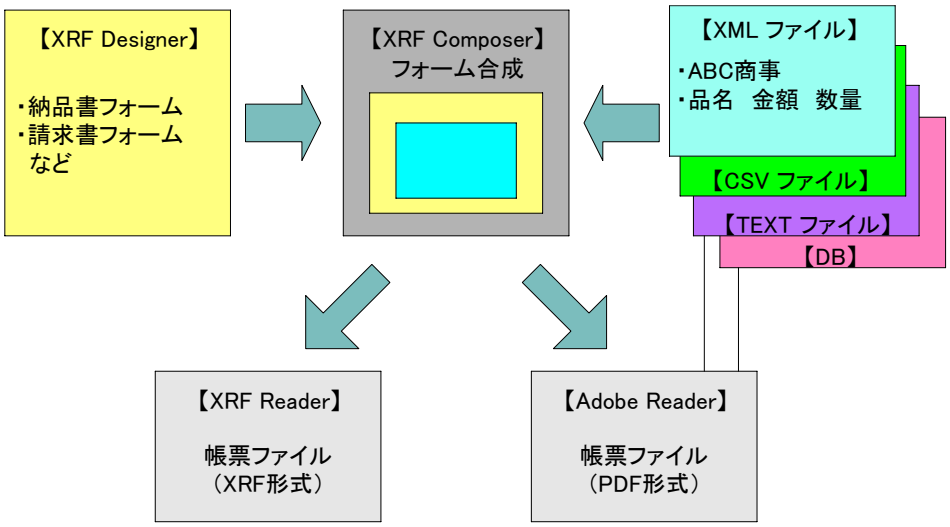
1. XRFの概要 1
1.1 XRFの特徴 1
1.2 XRFの構成 2
1.2.1 XRFのシステム構成図 2
1.2.2 XRFの構成要素 3
2. XRFの運用構成例 4
2.1 XRFの最小構成 4
2.2 XRFの中規模構成 5
2.3 XRFの大規模構成 6
3. 対応可能なフォームと帳票の種類 7
3.1 対応可能なフォームの種類と形式 7
3.2 対応可能な帳票形式 11
4. 対応可能なデータの種類 15
4.1 データ定義体 15
4.2 サポートされるデータ定義体の形式 16
4.3 ユーザデータ 17
4.3.1 XMLデータ 17
4.3.2 CSVデータ 19
4.3.3 TEXTデータ 23
4.3.4 DBデータ（DB連携） 28
4.3.5 各エンコードがサポートする文字 29
4.4 フォントについて 30
4.4.1 外字の扱いについて 30
4.4.2 外国語・多言語の扱いについて 33
4.5 イメージデータについて 34
4.6 バーコード 34
5. 帳票生成 35
5.1 帳票生成の機能 35
5.1.1 帳票生成の動作 36
5.2 ダイナミックオーバレイの機能 37
5.2.1 ダイナミックオーバレイの動作 38
6. ジョブファイル 39
6.1 ジョブファイルの機能 39
6.1.1 ジョブファイル生成の動作 40
7. リモート処理（XRF Remote Launch Broker） 41
7.1 リモート処理（XRF Remote Launch Broker）の機能 41
7.1.1 リモート処理で帳票ファイルを分散印刷の動作 42
7.1.2 リモート処理で帳票生成から印刷までの動作 43
7.1.3 リモート処理でジョブファイルを印刷の動作 43
7.1.4 リモート処理でジョブファイル生成して印刷の動作 44
7.1.5 リモート処理で帳票生成、ジョブファイル生成して印刷実行の動作 45
8. 論理データ 46
8.1 データ構造と各データの説明 46

8.1.1 パターン例1 -- データフィールドだけがある場合 48
8.1.2 パターン例2 -- 固定テーブルが2つある場合 49
8.1.3 パターン例3 -- 2ページの場合 50
8.1.4 パターン例4 -- 可変テーブルの場合 51
8.1.5 パターン例5 -- 1ページをサイクリックする場合 52
8.1.6 パターン例6 -- 2ページをサイクリックする場合 53
8.1.7 パターン例7 -- 可変テーブルがあるものをサイクリックする場合 54
8.1.8 パターン例8 -- 複写帳票でフィールドしかない場合 55
8.1.9 パターン例9 -- 複写帳票でテーブルがある場合 56
8.1.10 パターン例10 -- 複写帳票をサイクリックする場合 57
8.1.11 パターン例11 -- テーブルがある複写帳票をサイクリックする場合 58
索 引 59

1.　XRFの概要

XRFは、どのようなレイアウトの帳票でも自由に設計できます。XRF Designerで帳票フォームを作成し、XRF Composerで帳票フォームとお客様のもつユーザデータを重ね合わせることで帳票ファイルを作成します。作成した帳票ファイルは、XRF ReaderまたはAdobe Readerで表示・印刷することができます。

XRFは、帳票を作成したり、表示、印刷したりできるツールです。



1.1　XRFの特徴

XRFには、以下の特徴があります。

- **生産性の高いフォーム作成**

  XRF Designerを利用することで、複雑な構造を意識することなくフォームを作成でき開発工数を劇的に削減できます。

- **データ定義の高生産性**

  XRF Designerで行うデータマッピングはプログラムレスです。

  XMLデータをGUI上で直接マッピングすることができ、Oracle、SQL Server、DB2などのDBへの関連付けもできます。

- **多彩なデータ取り込み**

  XRF Composerは、多彩なデータ取り込みで帳票生成を柔軟に対応します。データ取り込みできるファイルは、XML、CSV、TEXT形式の固定長または可変長です。また、データベース（Oracle、SQL Server、DB2）からも取り込めます。

- **データ量に応じて明細行を加減**

  一覧形式や明細のある帳票は、データ量に応じて明細行を自動的に加減します。

- **多国語対応**

  XRF Composerは、対応文字コードとしてUTF-8をサポートしているため、中国語、韓国語等が容易に印字できます。

- **EXCELなどで分析、編集**

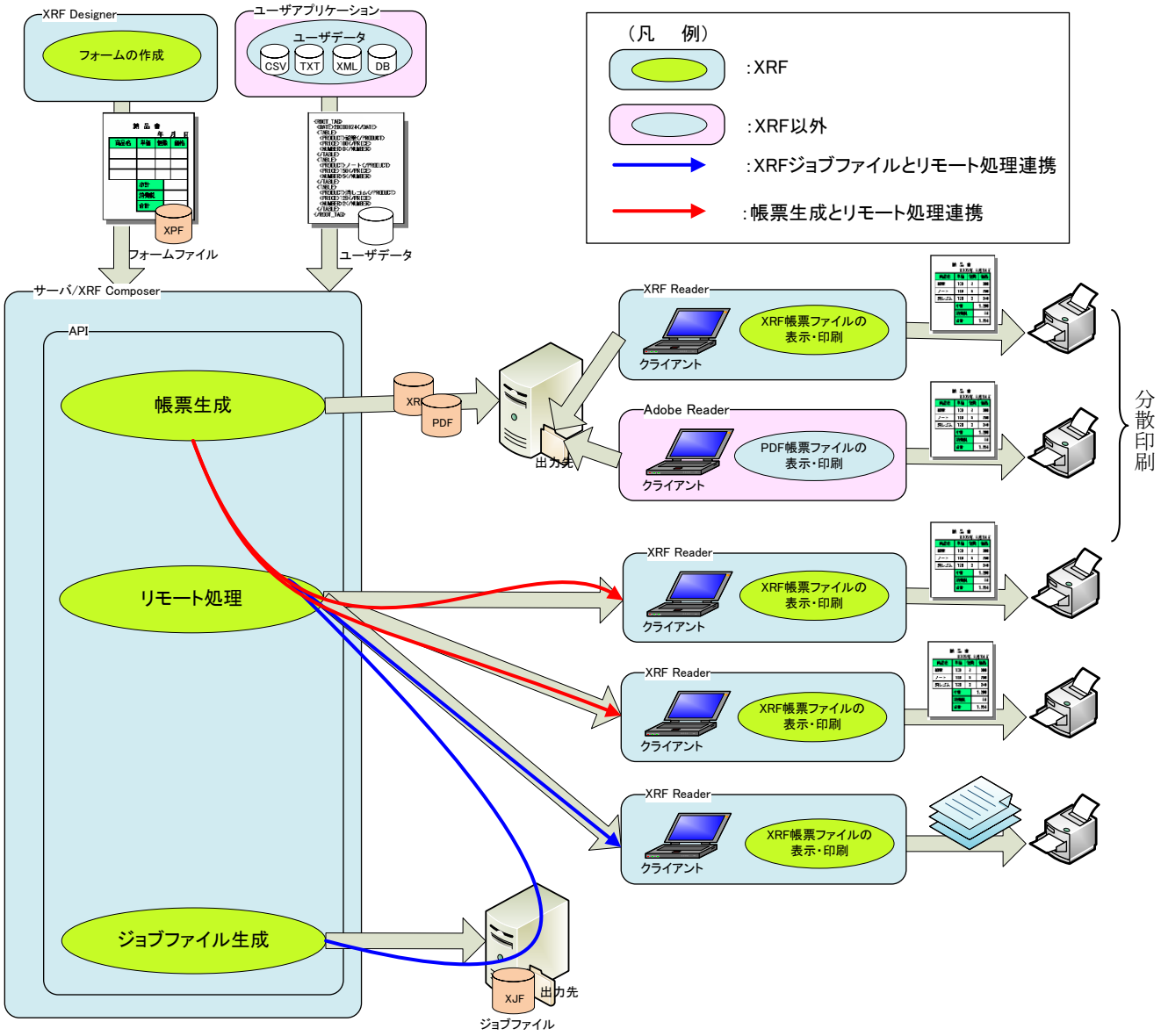
  XRF Readerは、２次加工用に帳票データのXML形式による抽出が可能です。抽出したXML形式のデータをEXCELなどで再利用し活用することができます。

1.2　XRFの構成

XRFの構成について説明します。

1.2.1　XRFのシステム構成図

XRFを利用する際のシステム構成は以下のとおりです。



※Adobe Readerは、XRFに含まれません。

1.2.2　XRFの構成要素

XRFは以下のアプリケーションから構成されています。

- XRF Designer

  XRF Designerは、帳票のフォームを作成するためのGUIツールです。罫線やタイトルといった図形を作成したり、ユーザデータを表示する場所や表示属性を指定するデータフィールドを作成したりします。ユーザデータのデータフォーマット（データ定義体）の規定やDBとの連携方法の設定もここで行います。

- XRF Viewer

  XRF Viewerは、XRF Designerで設計したフォームにユーザデータを貼り合わせて帳票として表示したり印刷したりするためのツールです。

- XRF Composer

  XRF Composerは、XRF Designerで作成したXPFフォームとお客様の持つユーザデータを組み合わせ、XRF形式または、PDF形式の帳票ファイルを生成します。また、複数のXRFファイルを1つのファイルにまとめるジョブファイルを生成することができます。

  XRF Composerは、Javaで開発されたライブラリです。ユーザアプリケーションまたはコマンドラインから、XRF ComposerのAPIを呼び出して帳票ファイルやジョブファイルを生成します。

- XRF Reader

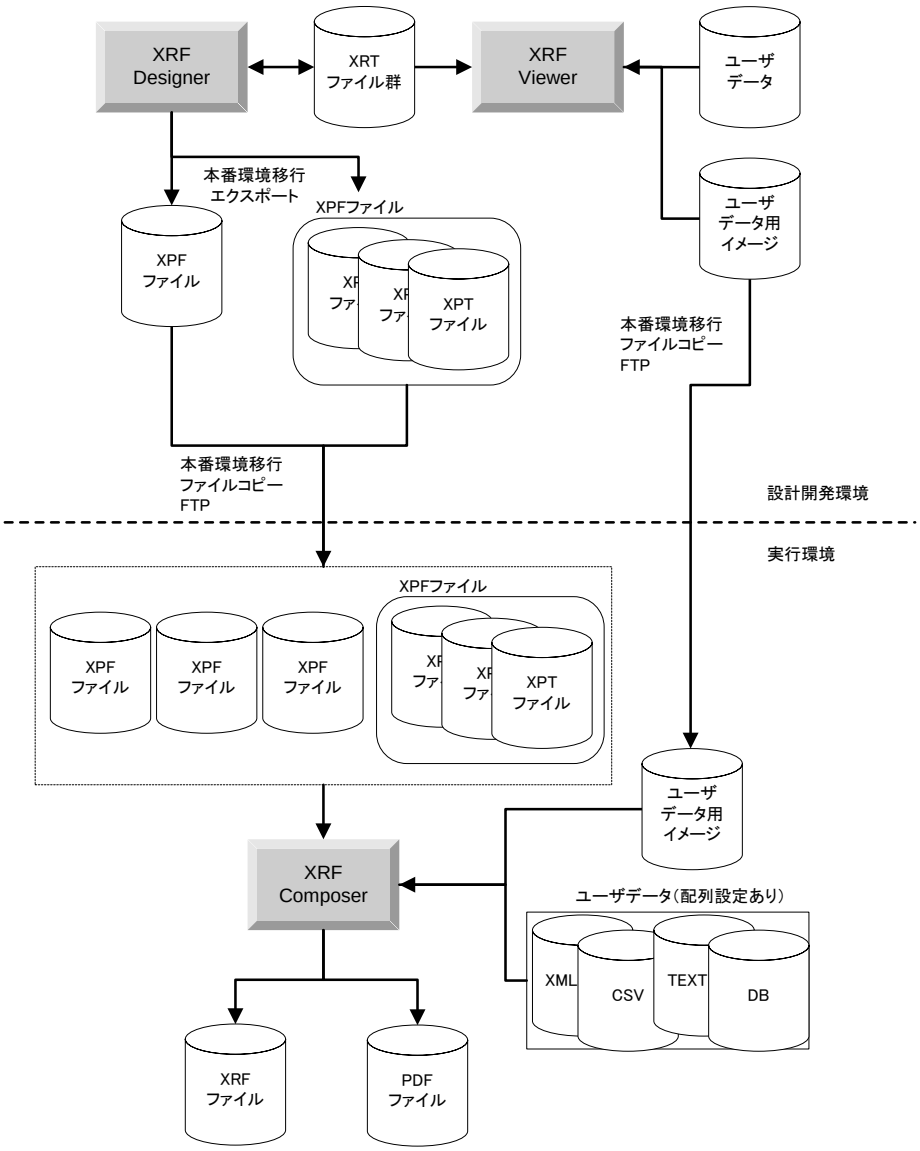
  XRF Readerは、XRF形式の帳票ファイルを表示、印刷および、帳票に含まれるフィールドを検索するためのツールです。複数フィールドを組み合わせた「複合検索」や、データ入力可能なフィールドにXRF Readerでデータを入力することもできます。リモート処理を実行する際は、XRF Readerを常駐させる必要があります。また、XRF Readerを常駐させることにより、帳票印刷を繰り返し実行する環境では、起動時間を短縮して印刷の効率を高めることができます。

2.　XRFの運用構成例

XRFを使った帳票システムの構成例を運用規模別に説明します。

2.1　XRFの最小構成

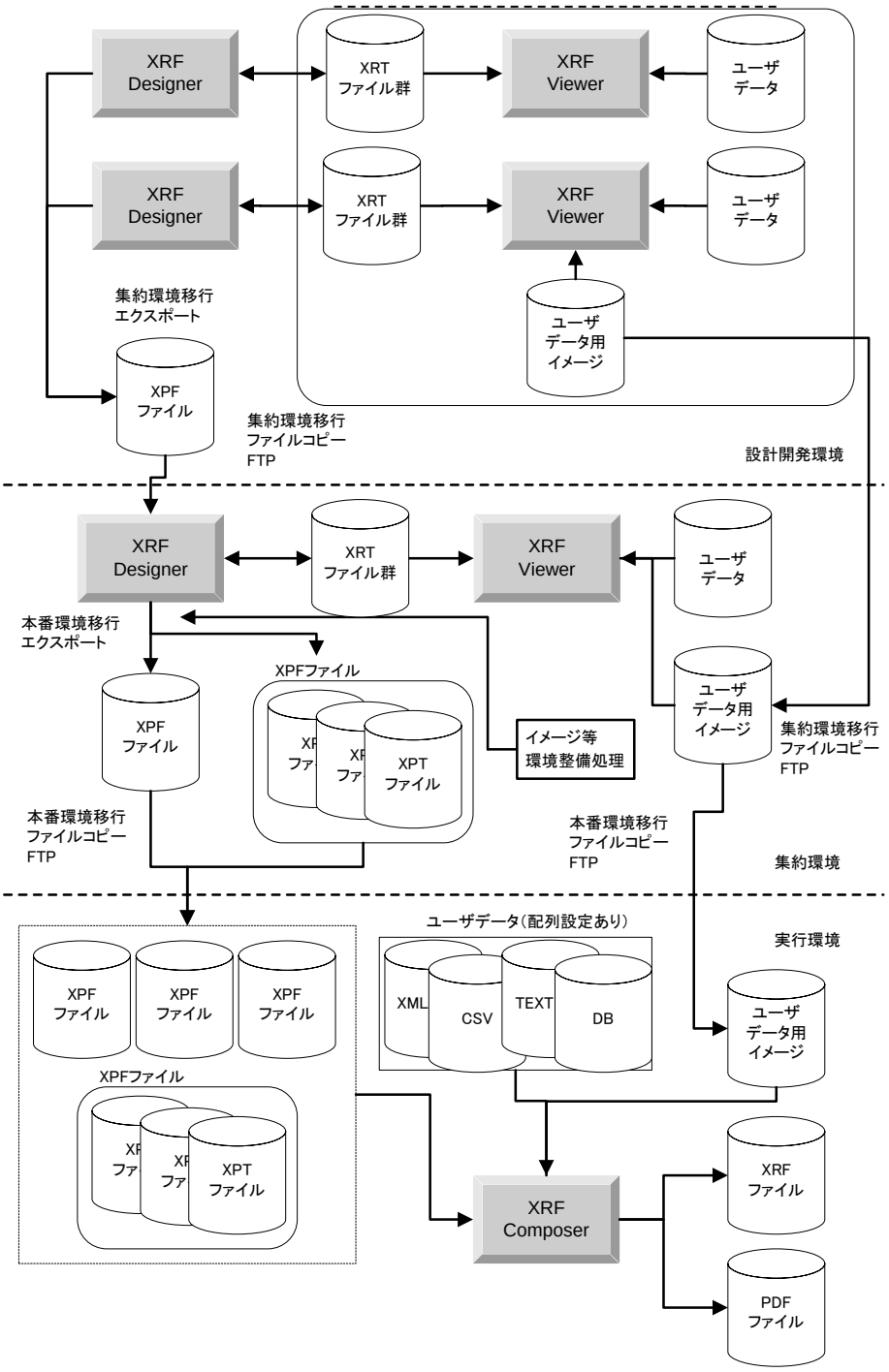
フォーム作成担当者が1名の構成例で、XRFを使う場合の最小構成となります。フォーム作成担当者はXRF Designerを使ってフォームを作成し、できあがったフォームをXRF Composerがインストールされた環境にコピーします。



運用構成図（最小構成）

2.2　XRFの中規模構成

フォーム作成担当者が2～3名いる場合の構成例です。各フォーム作成担当者は、自分の担当したフォームを作成し、XRF Composerがインストールされた環境にそれぞれコピーします。作成したフォームの管理はフォームのコピー先で行います。



運用構成図（中規模構成）

2.3　XRFの大規模構成

フォーム作成担当者が大勢いる場合の構成例です。各フォーム作成者が作成したフォームは、直接XRF Composerがインストールされた環境にコピーせず、集約用のXRF Designerがインストールされた環境にコピーします。フォームの管理者がコピーされたフォームの中から必要なものだけをまとめてXRF Composerがインストールされた環境へとコピーします。

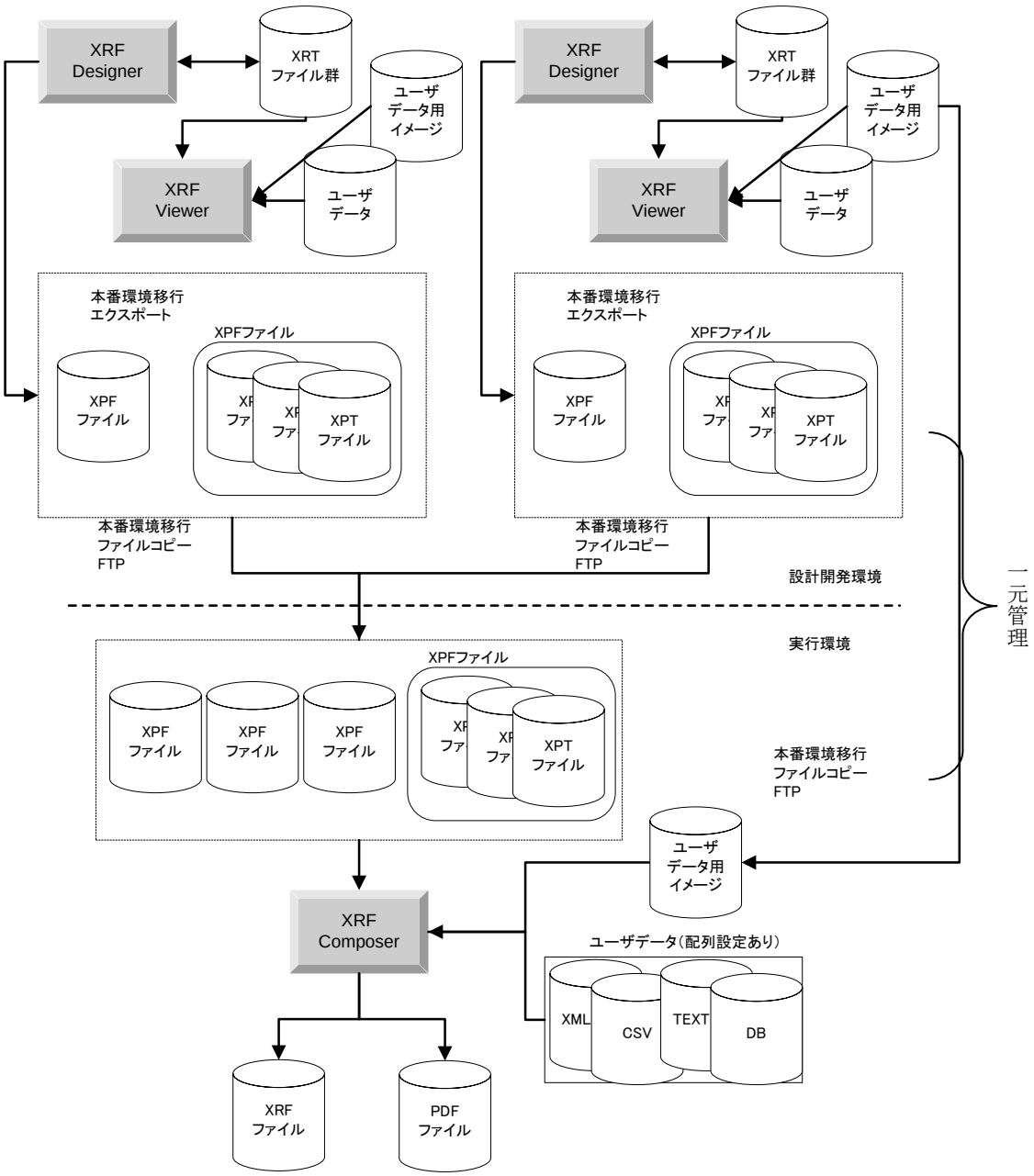


図 1 運用構成図（大規模構成）

3.　対応可能なフォームと帳票の種類

ここでは、XRFで作成できるフォームの種類と表現可能な帳票の例を説明します。

3.1　対応可能なフォームの種類と形式

XRFでは、以下のフォーム種類と形式を取り扱うことができます。

＜種類＞

- シンプルフォーム
- トレースフォーム

＜形式＞

- サイクリック形式
- ロット化形式
- ラインデータフォーム

● シンプルフォーム

単純な1ページの様式、連続帳票様式および複数ページの様式の作成が可能です。

以下の例はいずれもシンプルフォームの例です。

納 品 書

20XX年 9月24日

商品名	単価	個数	価格
鉛筆	100	3	300
ノート	150	5	750
消しゴム	120	2	240
	小計		1,290
	消費税		64
	合計		1,354

1ページのみのフォーム例

1ページ目

納 品 書

20XX年 9月24日

商品名	単価	個数	価格
鉛筆	100	3	300
ノート	150	5	750
消しゴム	120	2	240
万年筆	650	2	1,300
三角定規	240	4	960

2ページ目以降

商品名	単価	個数	価格
分度器	320	1	320
コンパス	300	3	900
絵の具	560	4	2,240
筆	180	10	1,800
色鉛筆	720	2	1,440
墨汁	500	3	1,500
画用紙	350	5	1,750

最終ページ

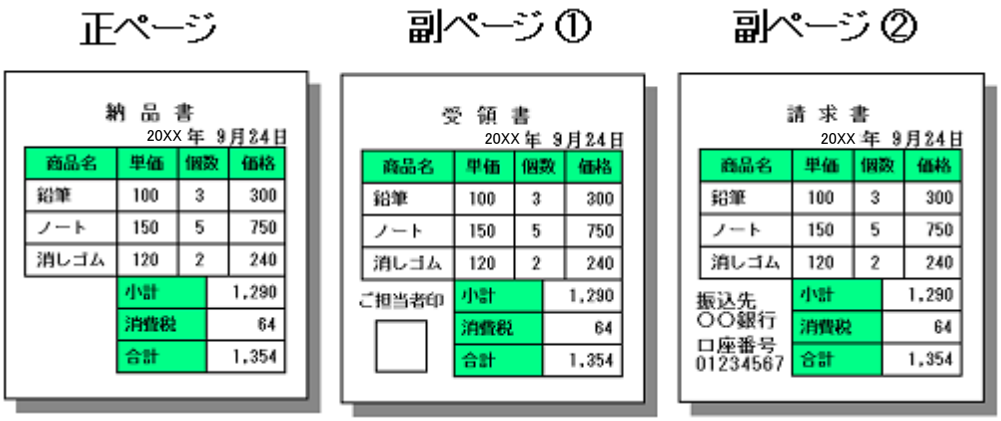
商品名	単価	個数	価格
はさみ	380	2	760
クレヨン	920	1	920
	小計		15,180
	消費税		759
	合計		15,939

ユーザデータ量に合わせて、改ページしながら複数ページ出力するフォーム例

- シンプルフォームの作成手順については、『XRF Designer 操作マニュアル』の『3.2　シンプルフォームを作成する』を参照してください。

- **トレースフォーム**

  複写伝票のように、1つの『正ページ』と重なっている1つ以上の『副ページ』から構成されるフォームです。それぞれのページに共通する図形やデータフィールドを一度に作成することができます。複写伝票は基本的に同じデータフィールドを持ちますので、後からコピーするのではなく、一度に作成できるので開発効率が向上します。不要なものはマスクしたり、データフィールドの属性を変更したりして非表示にすることができます。



複写伝票の正、副、控のように、1つのユーザデータで

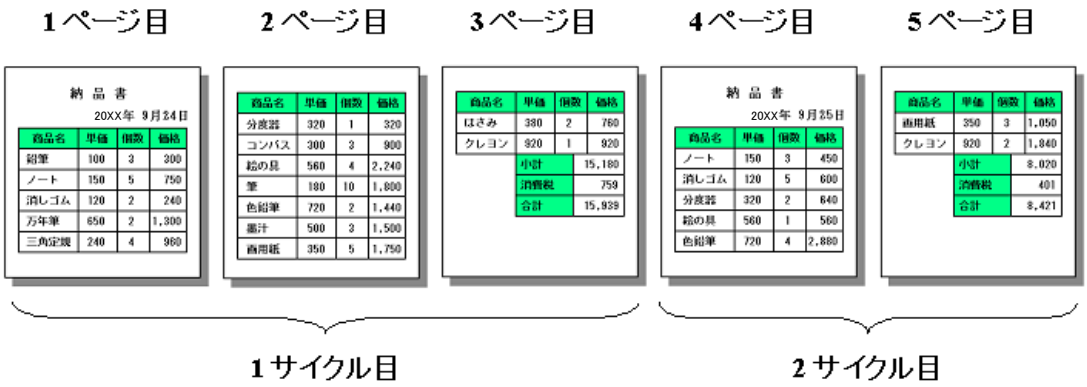
複数ページの異なる様式で出力するフォーム例

- 『正ページ』が複数あるトレースフォームは作成できません。

- トレースフォームの作成手順については、『XRF Designer 操作マニュアル』の『3.3　トレースフォームを作成する』を参照してください。

● **サイクリック形式**

シンプルフォームやトレースフォームが単純に繰り返している形式です。それぞれのフォーム形式で作成したフォームに対してサイクリックの指定を行います。



サイクリックの例

- サイクリック形式の作成手順については、『XRF Designer 操作マニュアル』の『4.　サイクリックフォームを作成する』を参照してください。

● **ロット化形式**

複数のフォームを作成し、帳票ファイルの生成時に1つの帳票ファイルにする形式です。シンプルフォームやトレースフォーム、およびこれらをサイクリック指定にしただけでは表現できない帳票の場合にこのロット化形式を使用します。ロット化形式にまとめられるフォームの数は最大10フォームまでです。尚、ページ番号は、各XRT内でのみページ番号をカウントアップするか、複数のXRTをまたがったページ番号をカウントアップします。

ロット化の例

- ロット化とは、XRF Designerで作成した複数のXRTフォームファイルを、ひとつのXPFフォームファイルにまとめる機能です。
- ロット化の作成手順については、『XRF Designer 操作マニュアル』の『3.5.　ロット化フォームを作成する』を参照してください。

- **ラインデータフォーム**

  ラインデータ用のフォームです。ページサイズやグリッドの設定を元に、ラインデータ用フォームのフィールドを作成、配置を自動で行います。更に、作成したフィールドに対応するデータ定義体を同時に作成します。複数のライン状ユーザフィールドを連続させるフラットタイプと、リンクテーブルを使用するテーブルタイプがあります。ラインデータの長さに合わせてページとグリッドの設定を変更し、定義体を生成してください。

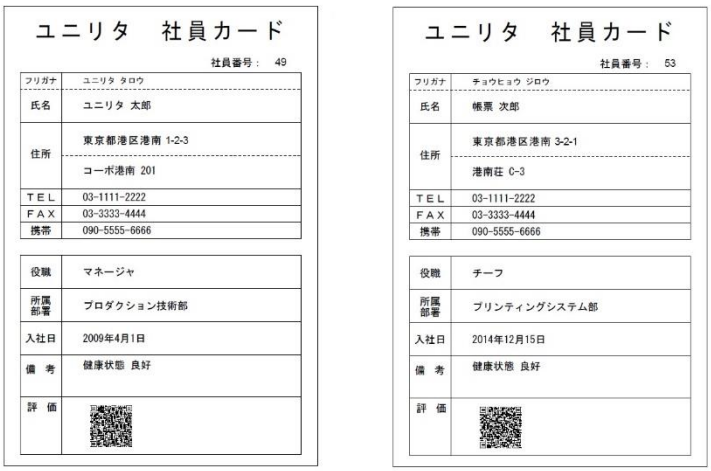
> ラインデータフォームの作成手順については『XRF Designer操作マニュアル』『 3.6 ラインデータ用フォームの作成』を参照してください。

3.2　対応可能な帳票形式

XRFを使って表現できる帳票の例を説明します。

- **カード形式帳票**

行数が可変となるデータを含まず、固定ページ数で内容が完結する帳票形式です。



ユニリタ　社員カード

社員番号：　49

フリガナ	ユニリタ タロウ
氏名	ユニリタ 太郎
住所	東京都港区港南 1-2-3
	コーポ港南 201
ＴＥＬ	03-1111-2222
ＦＡＸ	03-3333-4444
携帯	090-5555-6666

役職	マネージャ
所属部署	プロダクション技術部
入社日	2009年4月1日
備　考	健康状態 良好
評　価	

ユニリタ　社員カード

社員番号：　53

フリガナ	チョウヒョウ ジロウ
氏名	帳票 次郎
住所	東京都港区港南 3-2-1
	港南荘 0-3
ＴＥＬ	03-1111-2222
ＦＡＸ	03-3333-4444
携帯	090-5555-6666

役職	チーフ
所属部署	プリンティングシステム部
入社日	2014年12月15日
備　考	健康状態 良好
評　価	

カード形式帳票の例

- **伝票形式帳票**

可変行数のデータを含みますが、1ページで内容が完結する帳票形式です。

見積書No.：103-23-77
2009年2月17日

御見積書

株式会社グローバライン　御中

下記のとおりお見積申し上げます。
合計金額：¥7,000,000
納　　期：水曜～翌月曜日発送
取引方法：納品月末～翌25日現金払い
有効期限：ご提示後1ケ月

〒108-6029
東京都港区港南
株式会社ユニリタ
担当：ユニリタ 太郎
E-Mail：sales@unirita.co.jp
TEL：03-5463-9881
FAX：03-5463-4760

製品名	価格	数量	金額
DURL Windows版	¥490,000	10	¥4,900,000
DURL 追加クライアントライセンス	¥150,000	10	¥1,500,000
FormHelper	¥198,000	1	¥198,000
ROF Composer	¥800,000	1	¥800,000
ROF Reader	¥0	1	¥0
	小　計		¥7,398,000
	特別値引		▲ ¥398,000
	合　計		¥7,000,000

伝票形式帳票の例

- **伝票（複数ページ）形式帳票**

  可変行数のデータを含み、更に1ページで内容が完結しない帳票形式です。鏡ページ（1ページ目）と明細ページ（2ページ目以降）とで異なるフォームを使うことができます。

見積書No.：103-23-77
2009年2月17日

御見積書

株式会社グローバライン　御中

下記のとおりお見積申し上げます。
合計金額：¥20,000,000
納　　期：水曜〆翌月曜日発送
取引方法：納品月末〆翌25日現金払い
有効期限：ご提示後1ケ月

〒108-6029
東京都港区港南
株式会社ユニリタ
担当：ユニリタ 太郎
E-Mail：sales@unirita.co.jp
TEL：03-5463-9881
FAX：03-5463-4760

製品名	価格	数量	金額
DURL Windows版	¥490,000	10	¥4,900,000
DURL Solaris版	¥800,000	10	¥8,000,000
DURL 追加クライアントライセンス	¥150,000	2	¥300,000
DURL 追加プリンタライセンス	¥50,000	100	¥5,000,000
FormHelper	¥198,000	1	¥198,000
ROF Composer	¥800,000	1	¥800,000
ROF Reader	¥0	1	¥0

製品名	価格	数量	金額
DURL トレーニング	¥200,000	1	¥200,000
FormHelper トレーニング	¥200,000	1	¥200,000
SE導入支援	¥1,000,000	1	¥1,000,000
オーバーレイ変換サービス	¥5,000	50	¥250,000
オーバーレイ作成サービス	¥30,000	20	¥600,000
	小　計		¥21,448,000
	特別値引		▲ ¥1,448,000
	合　計		¥20,000,000

伝票（複数ページ）形式帳票の例

- **伝票（複数領域）形式帳票**

可変行数のデータを含み、データが最初のテーブルを溢れた場合、次のテーブルに続きを表示する帳票形式です。

請求書

お得意様名
東京都港区港北 4-5-6
港北ビルディング 30F
得意 先夫　御中

〒108-6029
東京都港区港南
株式会社ユニリタ

期間	合計	消費税	請求金額
09/02/03 ～ 09/03/31	¥866,500	¥43,325	¥909,825

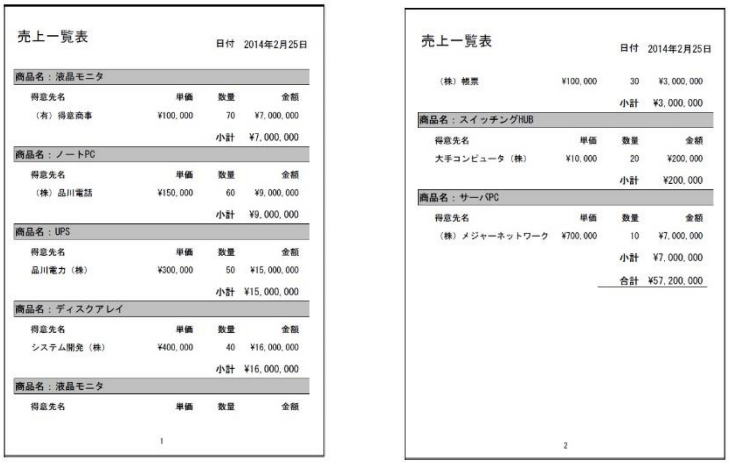
月	日	伝票	品名	数量	単価	金額
3	3	101	CDケース	10	1,000	10,000
			インクリボン	5	1,500	7,500
			LANケーブル	10	2,000	20,000
			3月3日 小計		3 件	37,500
3	10	102	HUB	1	7,000	7,000
			メモリ	5	7,500	37,500
		102	ハードディスク	10	8,000	80,000
		103	キーボード	2	9,000	18,000
			マウス	25	10,000	250,000
			3月10日 小計		5 件	392,500
3	25	104	CDケース	10	1,000	10,000
			インクリボン	5	1,500	7,500
		105	LANケーブル	10	2,000	20,000
			CDケース	20	1,000	20,000
			インクリボン	20	2,000	40,000
		106	DDRメモリ	20	3,000	60,000
			テーブルタップ	20	4,000	80,000
			ノートPCケース	20	5,000	100,000
			DDRメモリ	15	3,000	45,000
			3月25日 小計		9 件	382,500

月	日	伝票	品名	数量	単価	金額
3	31	201	マウス	1	10,000	10,000
		202	ハードディスク	3	8,000	24,000
		203	モニタ	1	20,000	20,000
			3月31日 小計		3 件	54,000

伝票（複数領域）形式帳票の例

- **一覧表形式帳票**

可変行数のデータを含み、データのグループ毎にタイトルや集計を表示する帳票形式です。



売上一覧表　日付 2014年2月25日

商品名：液晶モニタ

得意先名	単価	数量	金額
（有）得意商事	¥100,000	70	¥7,000,000
		小計	¥7,000,000

商品名：ノートPC

得意先名	単価	数量	金額
（株）品川電話	¥150,000	60	¥9,000,000
		小計	¥9,000,000

商品名：UPS

得意先名	単価	数量	金額
品川電力（株）	¥300,000	50	¥15,000,000
		小計	¥15,000,000

商品名：ディスクアレイ

得意先名	単価	数量	金額
システム開発（株）	¥400,000	40	¥16,000,000
		小計	¥16,000,000

商品名：液晶モニタ

得意先名	単価	数量	金額

1

売上一覧表　日付 2014年2月25日

（株）帳票	¥100,000	30	¥3,000,000
		小計	¥3,000,000

商品名：スイッチングHUB

得意先名	単価	数量	金額
大手コンピュータ（株）	¥10,000	20	¥200,000
		小計	¥200,000

商品名：サーバPC

得意先名	単価	数量	金額
（株）メジャーネットワーク	¥700,000	10	¥7,000,000
		小計	¥7,000,000
		合計	¥57,200,000

2

一覧表形式帳票の例

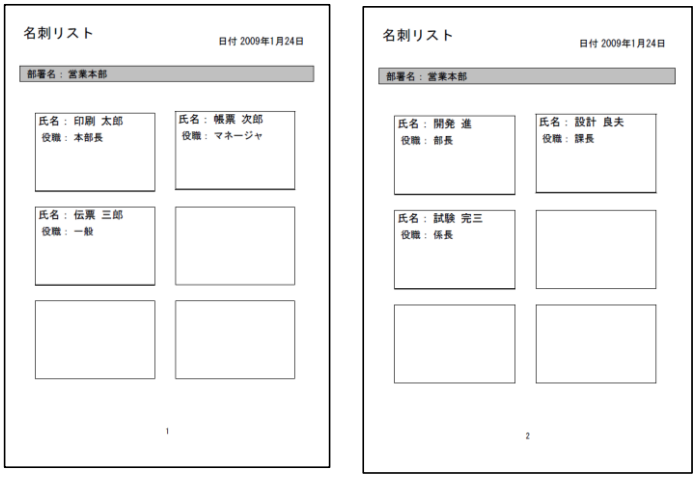
- **ラベル形式帳票**

  1ページ内に複数枚のラベルを表示する帳票形式です。データの量に応じて自動的に改ページします。

ラベル形式帳票の例

- **ラベル（ラベル外データ付き）形式帳票**

  ラベル形式帳票の特徴に加え、ラベルの他にも区分け用のデータが表示される帳票形式です。データの量が多かったり、区分け用のデータが切り替わったりしたタイミングで自動的に改ページします。



ラベル（ラベル外データ付き）形式帳票の例

4.　対応可能なデータの種類

ここでは、扱えるデータ（データ定義体、ユーザデータ、フォント、外字、イメージ、バーコード）の種類について説明します。

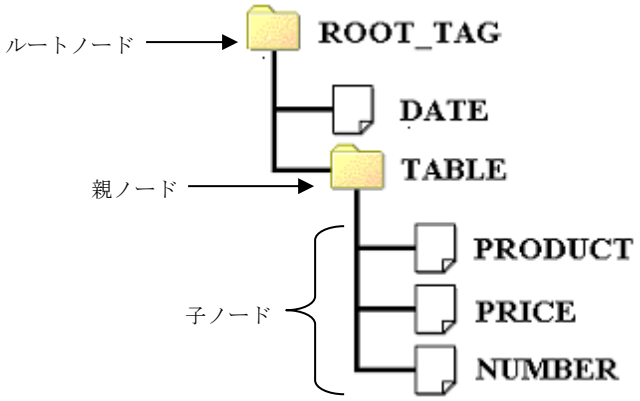
4.1　データ定義体

XRFでは、ユーザデータを『データ定義体』によって意味付けすることで、ページやテーブル（表）の明細部分といった繰り返し部分を判断したり、どの項目をどのデータフィールドに表示するかといったマッピング（フォームとユーザデータとの紐付け）に利用したりします。データ定義体はフォームの作成時に新たに作ることも、既存のものを再利用することもできます。

データ定義体はデータ項目を表す子ノードと、ノードの集まりを表す親ノードとを使って階層的に表現されます。それぞれのノードには発生回数を指定します。発生回数は、『任意』、『0以上（繰り返し）』のいずれかより選択します。ただし、子ノードに対して繰り返しを指定した場合、ユーザデータ中で最初に出現したノードだけが使用されて残りは無視されます。また、親ノードに対して繰り返し以外の出現回数を指定することはできません。

子ノードは対応するデータ項目のデータ型を持っています。使用できるデータ型は『数値型』、『文字型』および『日付型』の3種類です。

親ノードは配下のノードの繰り返しを意味します。また、親ノードの下に親ノードがあることもあります。全てのノードの一番上には、ルートノードがあります。



データ定義体のイメージ

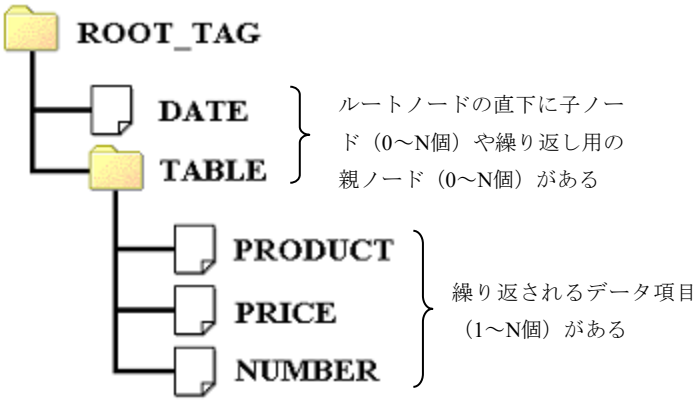
- データ定義体の作成手順については、『XRF Designer 操作マニュアル』の『2.　ユーザデータマッピング』、『3.2.1.5　データ定義体の自動作成とマッピング』を参照してください。

4.2　サポートされるデータ定義体の形式

XRFでサポートされるデータ定義体の形式は次の2通りです。

- **フラットパターン**

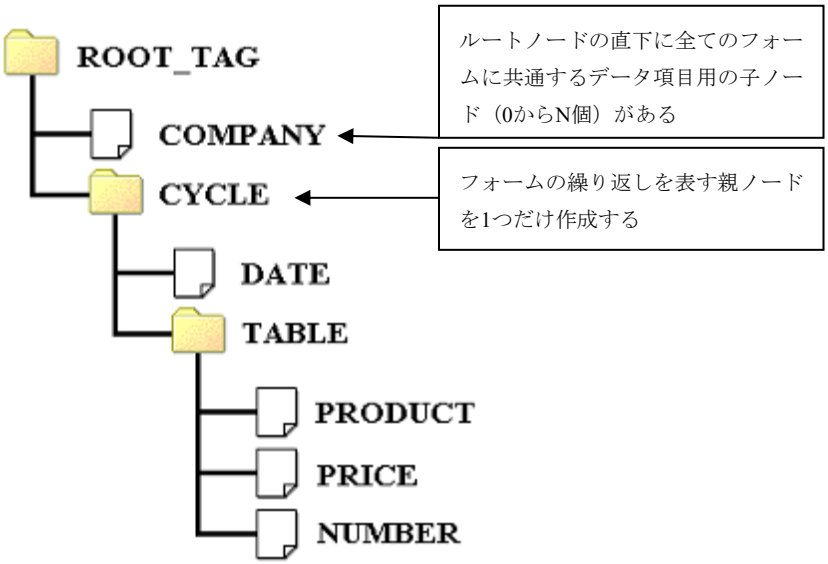
  サイクリック以外のフォームのためのパターンです。ルートノードの直下に子ノードや、テーブルやグラフといった繰り返しデータが必要なオブジェクト用の親ノードがあります。



データ定義体（フラットパターン）

- **サイクリックパターン**

  サイクリック用のパターンです。ルートノードの直下にフォームの繰り返しを表す親ノードが1つだけあり、その直下に子ノードや、テーブルやグラフといった繰り返しデータが必要なオブジェクト用の親ノードがあります。ルートノードの直下に、全てのサイクルに共通するデータ用の子ノードを置くこともできます。



データ定義体（サイクリックパターン）

4.3　ユーザデータ

ユーザデータはユーザアプリケーションで生成され、データフィールドに当てはめるデータのことです。ここではユーザデータに関して説明します。

- ユーザデータの作成手順については、『XRF Designer 操作マニュアル』の『3.2.1.6.　ユーザデータ（XML形式）サンプルの作成』を参照してください。

4.3.1　XMLデータ

『4.1　データ定義体』で解説した構造をもつXMLデータをサポートします。XRFで扱うことのできるXMLデータはW3Cの勧告（1.0）全てを網羅するわけではありませんのでご注意ください。例えば各エレメントに属性を指定しても無視されます。

- 対応するエンコードは次のとおりです。

  UTF-8

  UTF-16

  Shift-JIS（MS932）

  EUC-JP

```
<?xml version="1.0" encoding="UTF-8"?>
<ROOT_TAG>
 <DATE>20030924</DATE>
 <TABLE>
  <PRODUCT>鉛筆</PRODUCT>
  <PRICE>100</PRICE>
  <NUMBER>3</NUMBER>
 </TABLE>
 <TABLE>
  <PRODUCT>ノート</PRODUCT>
  <PRICE>150</PRICE>
  <NUMBER>5</NUMBER>
 </TABLE>
 <TABLE>
  <PRODUCT>消しゴム</PRODUCT>
  <PRICE>120</PRICE>
  <NUMBER>2</NUMBER>
 </TABLE>
</ROOT_TAG>
```

※ UTF-8の場合

XMLデータの例

4.3.1.1　XMLデータ/エンコードの指定方法

XML形式のユーザデータにおいてエンコードの指定は、以下のとおりです。

- XMLデータの1行目で指定する

● XMLデータの1行目でエンコードを指定する

ここでは、XMLデータの1行目でエンコードを指定する方法について説明します。

指定可能なエンコードは、以下のとおりです。

エンコード	記述例
UTF-8	<?xml version=”1.0” encoding=”UTF-8” ?>
UTF-16	<?xml version=”1.0” encoding=”UTF-16” ?>
Shift-JIS	<?xml version=”1.0” encoding=”Shift-JIS” ?>
EUC	<?xml version=”1.0” encoding=”EUC-JP” ?>
MS932	<?xml version=”1.0” encoding=” MS932” ?> <?xml version=”1.0” encoding=” Windows-31J” ?>

● 省略時の動作

XMLデータにおいてエンコードの指定は省略可能です。省略した場合、UTF-8が有効となります。

4.3.2　CSVデータ

CSVデータはフラットなデータですが、XMLデータと同様にデータ定義体を利用して繰り返しデータを指定することができます。尚、XRFで扱うことのできるCSVデータは次の条件を満たす必要があります。

- 対応するエンコードは次のとおりです。
  Shift-JIS
  MS932
  EUC_JP
  UTF-8
  UTF-16 *
  * UTF-16は「設定ファイル」にて指定可能です。「CSVデータの1行目」では指定できません。
- レコードは改行、フィールドは半角カンマ（,）で区切る必要があります。
- 囲み文字（"）でフィールドデータを囲むことも可能です。
- 囲み文字（"）をデータとして使用する場合は引用符（"）を2つ続けて記述します。
  半角カンマ（,）を区切り文字ではなくデータとして使用する場合は、フィールドデータを引用符（"）で囲みます。
- FormHelper型CSVで使われているページコードを含んだCSVデータは対象外です。
  *FormHelperは、弊社の製品です。
- ヘッダレコードが必要です。（データ定義体の子ノード名に対応するフィールド名が指定されている必要があります）

```
DATE,     PRODUCT, PRICE, NUMBER
20030924, 鉛筆,     100,     3
20030924, ノート,   150,     5
20030924, 消しゴム, 120,     2
```

CSVデータの例

4.3.2.1　CSVデータ/エンコードの指定方法

CSV形式のユーザデータにおいてエンコードの指定には、以下の方法があります。

- CSVデータの1行目で指定する
- コンフィグ設定ファイルで指定する

● CSVデータの1行目でエンコードを指定する

ここでは、CSVデータの1行目でエンコードを指定する方法について説明します。

【書式】1行目に以下を指定します。
　CSVENCODING=*キーワード*

指定可能なキーワードは以下のとおりです。（大文字、小文字は区別しません。）

Shift-JISを指定する場合は、以下のいずれかを指定します。

SJIS
Shift-JIS
Shift_JIS

MS932を指定する場合は、以下のいずれかを指定します。

MS932
Windows-31J

EUC_JPを指定する場合は、以下のいずれかを指定します。

EUC-JP
EUC_JP

UTF-8を指定する場合は、以下を指定します。

UTF-8

【記述例】

```
CSVENCODING=SJIS
DATE,PRODUCT,UNIT PRICE,NUMBER
20050524,鉛筆,100,3
20050524,ノート,150,5
20050524,消しゴム,120,2
```

- 省略時の動作と優先順位

CSVデータにおいてエンコードの指定は省略可能です。省略した場合、コンフィグ設定ファイルの指定が有効となります。また、CSVデータとコンフィグ設定ファイルの両方でエンコードが指定されている場合、CSVファイルの指定が優先されます。

- XRF Composerのコンフィグ設定ファイルでエンコードを指定する

ここでは、XRF Composerのコンフィグ設定ファイルでエンコードを指定する方法について説明します。

XRF Composerの「コンフィグ設定ファイル」は以下のファイルです。

XRF Composerのインストールディレクトリ/conf/conf.xml

【書式】以下のタグで指定します。

```
<CONFIG>
    <DEFAULT>
        <CSVENCODING>キーワード</CSVENCODING>
    </DEFAULT>
</CONFIG>
```

指定可能なキーワードは以下のとおりです。（大文字、小文字は区別しません。）

Shift-JISを指定する場合は、以下のいずれかを指定します。

SJIS
Shift-JIS
Shift_JIS

MS932を指定する場合は、以下のいずれかを指定します。

MS932
Windows-31J

EUC_JPを指定する場合は、以下のいずれかを指定します。

EUC-JP
EUC_JP

UTF-8を指定する場合は、以下を指定します。

UTF-8

UTF-16を指定する場合は、以下を指定します。

UTF-16

【記述例】

```
<CONFIG>
    <DEFAULT>
        <CSVENCODING>SJIS</CSVENCODING>
    </DEFAULT>
</CONFIG>
```

- 初期値と優先順位

コンフィグ設定ファイルにおいて<CSVENCODING>タグの指定は必須で、以下のように処理されます。

・タグ自体が省略されている場合、SJISとして扱う。

・タグはあるが、設定値が空または間違っている場合、エラーとなる。

また、CSVデータとコンフィグ設定ファイルの両方でエンコードが指定されている場合、CSVファイルでの指定が優先されます。

4.3.2.2　CSVデータがサポートするパターン

- **パターン**

サポートするパターンは、以下のとおりです。

1. サイクリック
2. テーブル（1つのみ。リンクも固定も可能）
3. グラフ（1つのみ。ただし、テーブルと同時に利用することはできない。）
4. 1と2（テーブルのあるサイクリック）、または、1と3の組み合わせ。（グラフのあるサイクリック）

シンプルとトレースについては、両方ともサポートします。

- **データ定義体構造**

サポートするデータ定義体構造は、以下のとおりです。

パターン①　繰り返しなし

```
root
        flat_element（1～N個）
```

パターン②　繰り返し1階層

```
root
        flat_element（1～N個）
        record*（1つのみ）
                rec_element（1～N個）
```

パターン③　繰り返し2階層

root

common_element（1～N個）

cycle*（1つのみ）

flat_element（1～N個）

record*（1つのみ）

rec_element（1～N個）

- **データのセット（繰り返しの判断）**

CSVでは、繰り返しの表現を「行」単位でしか記述できません。
したがって、XRFでは、繰り返し要素と同じレベルにあるフラットな要素（＝繰り返しでない要素）をブレーク条件として使用します。
（ここでは、ヘッダ行は処理対象から除外します。また、例としてcycleやrecordといった名前を使用します。）

パターン1）繰り返しなし

・ CSVの1行目のデータのみでXMLを生成する。2行目以降は無視する。

パターン2）繰り返し1階層

・ flat_elementには、1行目のデータがセットされる。
・ rec_elementには、各行のデータごとに<record> </record>で囲んで階層化してセットする。したがって、1行が1recordとなる。

パターン3）繰り返し2階層

・ common_elementには、1行目のデータがセットされる。
・ 前行の値と比べて、flat_elementのうち1つでも変化していたらブレークとして扱い、<cycle></cycle>で囲んで階層化してflat_elementとrecord（複数）をセットする。したがって、flat_elementのブレークごとに1cycleとなる。
・ 前行の値と比べて、flat_elementの値がすべて同じ場合、cycle内の継続として扱い、recordをセットする。
・ rec_elementには、各行のデータごとに<record> </record>で囲んで階層化してセットする。したがって、1行が1recordとなる。

- **項目の対応関係**

・ CSVで、ヘッダ行は必須です。1行目はヘッダ行とします。
・ CSVの項目名（タグ名）とデータ定義体の要素名の一致で判断します。（XRFのデータ定義体では、要素名はデータ定義体内でユニークである必要があります。）

- **エラー**

以下のケースはエラーとなります。

・ サポート外のデータ定義体パターンを使用している。
・ CSVが1行以下である。（0行：なにもない、1行：ヘッダのみ）
・ ヘッダまたはデータ行において、データ定義体で「必須」となっている要素名または要素が存在しない。

4.3.3　TEXTデータ

TEXTデータは、区切り文字により、データ（フィールド）やライン（行）、ページなどを構成するテキスト形式のデータです。主にラインプリンタ等で出力をおこなうデータを指します。

XRFで扱うことができるTEXTデータは下記のような条件があります。

- 対応するエンコードは下記のとおりです。

Shift-JIS

MS932

EUC_JP

UTF-8

UTF-16 *

* UTF-16は「設定ファイル」にて指定可能です。「TEXTデータの1行目」では指定できません。

関連づけられたデータ定義体が、ラインデータ用の定義体である場合は、ヘッダ行を含まないデータとして認識します。

4.3.3.1 TEXTデータ/エンコードの指定方法

TEXT形式のユーザデータにおいてエンコードの指定には、以下の方法があります。

・TEXTデータの1行目で指定する。(ラインデータ用データ定義体を使用していない場合)

・コンフィグ設定ファイルで設定する。

- TEXTデータの1行目でエンコードを指定する

ここでは、TEXTデータの1行目でエンコードを指定する方法について説明します。

【書式】1行目に以下を指定します。

ENCODE=キーワード

指定可能なキーワードは以下のとおりです。(大文字、小文字は区別しません。)

Shift-JISを指定する場合は、以下のいずれかを指定します。

SJIS

Shift-JIS

Shift_JIS

MS932を指定する場合は、以下のいずれかを指定します。

MS932

Windows-31J

EUC_JPを指定する場合は、以下のいずれかを指定します。

EUC-JP

EUC_JP

UTF-8を指定する場合は、以下を指定します。

UTF-8

UTF-16を指定する場合は、以下を指定します。

UTF-16

【記述例】

ENCODE=SJIS

DATE,PRODUCT,UNIT PRICE,NUMBER

20050524,鉛筆,100,3

20050524,ノート,150,5

20050524,消しゴム,120,2

- XRF Composerのコンフィグ設定ファイルでエンコードを指定する

ここでは、XRF Composer（別売）のコンフィグ設定ファイルでエンコードを指定する方法について説明します。

XRF Composerの「コンフィグ設定ファイル」は以下のファイルです。

XRF Composerのインストールディレクトリ/conf/conf.xml

【書式】以下のタグで指定します。

```
<CONFIG>
    <DEFAULT>
        <TEXT_DEFAULT>
            <TEXTENCODING>キーワード</TEXTENCODING>
        </TEXT_DEFAULT>
    </DEFAULT>
</CONFIG>
```

指定可能なキーワードは以下のとおりです。（大文字、小文字は区別しません。）

Shift-JISを指定する場合は、以下のいずれかを指定します。

SJIS

Shift-JIS

Shift_JIS

MS932を指定する場合は、以下のいずれかを指定します。

MS932

Windows-31J

EUC_JPを指定する場合は、以下のいずれかを指定します。

EUC-JP

EUC_JP

UTF-8を指定する場合は、以下を指定します。

UTF-8

UTF-16を指定する場合は、以下を指定します。

UTF-16

【記述例】

```
<CONFIG>
    <DEFAULT>
      <TEXT_DEFAULT>
        <TEXTENCODING>SJIS</TEXTENCODING>
      </TEXT_DEFAULT>
    </DEFAULT>
</CONFIG>
```

● 初期値と優先順位

コンフィグ設定ファイルにおいて<TEXTENCODING>タグの指定は必須で、以下のように処理されます。

・タグ自体が省略されている場合、SJISとして扱う。

・タグはあるが、設定値が空または間違っている場合、エラーとなる。

また、TEXTデータとコンフィグ設定ファイルの両方でエンコードが指定されている場合、TEXTファイルでの指定が優先されます。

4.3.3.2　TEXTがサポートするパターン

『4.3.2.2　CSVデータがサポートするパターン』を参照してください

● **データのセット（繰り返しの判断）**

TEXTでは、繰り返しの表現を「行」単位でしか記述できません。
したがって、XRFでは、繰り返し要素と同じレベルにあるフラットな要素（＝繰り返しでない要素）をブレーク条件として使用します。
（ここでは、ヘッダ行は処理対象から除外します。また、例としてcycleやrecordといった名前を使用します。）

パターン1）繰り返しなし

・ TEXTの1行目のデータのみでXMLを生成する。2行目以降は無視する。

パターン2）繰り返し1階層

・ flat_elementには、1行目のデータがセットされる。

・ rec_elementには、各行のデータごとに<record> </record>で囲んで階層化してセットする。したがって、1行が1recordとなる。

パターン3）繰り返し2階層

- common_elementには、1行目のデータがセットされる。
- 前行の値と比べて、flat_elementのうち1つでも変化していたらブレークとして扱い、<cycle> </cycle>で囲んで階層化してflat_elementとrecord（複数）をセットする。したがって、flat_elementのブレークごとに1cycleとなる。
- 前行の値と比べて、flat_elementの値がすべて同じ場合、cycle内の継続として扱い、recordをセットする。
- rec_elementには、各行のデータごとに<record> </record>で囲んで階層化してセットする。したがって、1行が1recordとなる。

● **項目の対応関係**

- TEXTは、データ定義体にLINEDATA_ROOT_TAGが指定されている場合はカラム行は不要です。
- TEXTはデータの１ライン目から順にデータ定義体の要素に（XRFのデータ定義体では、要素名はデータ定義体内でユニークである必要があります。）

● **エラー**

以下のケースはエラーとなります。

- TEXTが1行以下である。（0行：なにもない、1行：ヘッダのみ）
- ヘッダまたはデータ行において、データ定義体で「必須」となっている要素名または要素が存在しない。

4.3.3.3 デリミタコードについて

XRFは、TEXTのデータに付加されている区切り文字、捨て文字（不要な文字コード）を認識し、帳票を出力することができます。

「区切り文字」、「囲み文字」、「改行コード」、「改ページコード」、「捨て文字」に指定できるデリミタコードは以下のとおりです。

【区切り文字、囲み文字、改行コード、改ページコード】

文字	コード	文字	コード
SOH	0x01	STX	0x02
ETX	0x03	EOT	0x04
ENQ	0x05	ACK	0x06
HT	0x09	LF	0x0a
VT	0x0b	FF	0x0c
CR	0x0d	!	0x21
“	0x22	#	0x23
$	0x24	%	0x25
&	0x26	’	0x27
(	0x28	)	0x29
*	0x2a	,	0x2c
.	0x2e	/	0x2f
:	0x3a	;	0x3b
?	0x3f		

【捨て文字】

文字	コード	文字	コード
SOH	0x01	STX	0x02
ETX	0x03	EOT	0x04
ENQ	0x05	ACK	0x06
LF	0x0a	VT	0x0b
FF	0x0c	CR	0x0d
SO	0x0e	SI	0x0f
DLE	0x10	DC1	0x11
DC2	0x12	DC3	0x13
DC4	0x14	NAK	0x15
SYN	0x16	ETB	0x17
CAN	0x18	EM	0x19
SUB	0x1a	ESC	0x1b
FS	0x1c	GS	0x1d
RS	0x1e	US	0x1f
SP	0x20	!	0x21
"	0x22	#	0x23
$	0x24	%	0x25
&	0x26	’	0x27
(	0x28	)	0x29
*	0x2a	+	0x2b
,	0x2c	-	0x2d
.	0x2e	/	0x2f
0	0x30	1	0x31
2	0x32	3	0x33
4	0x34	5	0x35
6	0x36	7	0x37
8	0x38	9	0x39
:	0x3a	;	0x3b
<	0x3c	=	0x3d
>	0x3e	?	0x3f
@	0x40	A	0x41
B	0x42	C	0x43
D	0x44	E	0x45

文字	コード	文字	コード
F	0x46	G	0x47
H	0x48	I	0x49
J	0x4a	K	0x4b
L	0x4c	M	0x4d
N	0x4e	O	0x4f
P	0x50	Q	0x51
R	0x52	S	0x53
T	0x54	U	0x55
V	0x56	W	0x57
X	0x58	Y	0x59
Z	0x5a	[	0x5b
¥	0x5c	]	0x5d
^	0x5e	_	0x5f
`	0x60	a	0x61
b	0x62	c	0x63
d	0x64	e	0x65
f	0x66	g	0x67
h	0x68	i	0x69
j	0x6a	k	0x6b
l	0x6c	m	0x6d
n	0x6e	o	0x6f
p	0x70	q	0x71
r	0x72	s	0x73
t	0x74	u	0x75
v	0x76	w	0x77
x	0x78	y	0x79
z	0x7a	{	0x7b
|	0x7c	}	0x7d
~	0x7e		

4.3.4　DBデータ（DB連携）

XRFで動作確認を行ったDBMSは次の4つです。開発環境と運用環境とでDBMSを切替えて使用することができます。

Oracle 8i

Oracle 9i

SQL Server 2000

DB2 V8

- DBデータについては、『XRF Designer 操作マニュアル』の『10.　DBアクセス』を参照してください。

4.3.5　各エンコードがサポートする文字

各エンコードがサポートする文字は以下のとおりです。

- Shift-JIS

  JIS X 0201 ラテン文字・カタカナ

  JIS X 0208

- MS932

  JIS X 0201 ラテン文字・カタカナ

  JIS X 0208

  NEC特殊文字 13区

  NEC選定IBM拡張文字 89－92区

  IBM拡張文字 115－119区

  ユーザ定義外字 95－114区　0xF040－0xF9FC

- EUC_JP

  JIS X 0201 ラテン文字・カタカナ

  JIS X 0208

  JIS X 0212 JIS補助漢字

- UTF-8、UTF-16

  日本語でサポートする文字は、以下のとおりです。

  JIS X 0201 ラテン文字・カタカナ

  JIS X 0208

  JIS X 0212 JIS補助漢字

  JIS X 0213 JIS第3第4水準

- Unicodeへの変換

  XRF Composerでは、ユーザ定義外字を以下のUnicodeに変換します。

  MS932　0xF040－0xF9FC

  Unicode　U+E000－U+E757

4.4　フォントについて

ここでは、文字フォントについて説明します。

フォント設定ファイル（fontNamesInfo.xml）は、XRFシリーズ全てのアプリケーションの文字フォントに関する情報を管理します。

- 文字フォントの追加・変更を行った場合は、XRF全てのアプリケーションの同名ファイルを置き換えてください。
- フォント設定ファイルを編集する前に、バックアップを作成することをお勧めします。何らかの原因でフォント設定ファイルの記述を誤ると、XRFが起動しない、または正しく動作しないことがあります。このような場合に、バックアップと内容を比較して修正する、またはバックアップを使用して直前の状態に戻すことができます。
- 一度設定したNAME,ALIASの組み合わせは変更しないでください。

【記述説明】

例）

<FONT NAME="Courier New" ALIAS="Courier New" HALFPITCH="1229/1641" ASCENT="1255" DESCENT="386" />　※実際には1行で記述します。

NAME：実フォント名称

ALIAS：論理フォント名称（表示上のフォント名称）

ASCENT：フォント作成ツールで参照できる値を設定して下さい。

DESCENT：フォント作成ツールで参照できる値を設定して下さい。

HALFPITCH：半角フォントの縦横比（横/縦）を表します。なければプロポーショナルフォントとみなされます。

【フォント設定ファイル（fontNamesInfo.xml）】

アプリケーション	場所
XRF Composer	*XRF Composerのインストールディレクトリ*/conf/fontNamesInfo.xml
XRF Designer XRF Viewer	*XRF Designerのインストールディレクトリ*/conf/fontNamesInfo.xml
XRF Reader	*%ALLUSERSPROFILE%/XRF Reader*/conf/fontNamesInfo.xml

4.4.1　外字の扱いについて

ここでは、XRFで外字を使用する際の設定について説明します。

4.4.1.1　XRFとPDFにおける外字の扱い

XRFファイルでは、ご使用のOSに外字フォントがインストールされている必要があります。外字フォントを埋め込むことはできません。

PDFファイルでは、外字を埋め込む必要があります。外字を埋め込まなくてもPDFは生成できますが、外字は表示されません。

4.4.1.2　外字を使う場合のユーザデータの文字コード

XRFではユーザデータの文字コードとしてUTF-8、SJIS、 EUC、MS932をサポートしていますが、外字を含むデータの場合、XMLではUTF-8、CSV・TEXTではMS932形式が外字を表示することができます。

文字コードは、XRFおよびPDFファイルの生成において共通に必要な設定です。

- CSV、TEXT

外字を含むCSVデータ及びTEXTデータの文字コードは、ユーザデータの1行目または、コンフィグ設定ファイルでMS932を指定します。ユーザデータとコンフィグ設定ファイルの両方で文字コードが指定されている場合、ユーザデータ内の指定が優先されます。詳しくは『4.3.2　CSVデータ』『4.3.3　TEXTデータ』を参照してください。

- XML

外字を含むXMLデータの文字コードは、XMLデータの1行目でUTF-8を指定します。詳しくは『4.3.1 XMLデータ』を参照してください。

4.4.1.3　外字フォントの埋め込み

外字を埋め込んだPDF生成において、扱うことのできる外字フォントファイルは以下のとおりです。

- 外字フォント：TrueTypeフォント
- 外字フォントファイルの拡張子：.ttf
  外字フォントファイル名は任意につけることができます。

また、外字を埋め込んだPDFを生成する場合、外字フォントファイルの保存ディレクトリを指定する必要があります。保存ディレクトリの指定は、PDF生成時の設定パラメータ、または、コンフィグ設定ファイルで以下のとおりに指定します。

- **設定パラメータで指定**

PDF生成時に外字ファイルの保存ディレクトリを指定するには、以下のパラメータを使用します。

PDF_EXT_FONT(pef)

パラメータについては、『XRF Composer 開発ガイド』を参照してください。

- **コンフィグ設定ファイルで指定**

コンフィグ設定ファイル（*XRF Composerのインストールディレクトリ*/conf/conf.xml）では外字ファイルの保存ディレクトリを以下のタグで指定します。

形式

<PDF_EXT_FONT>*保存ディレクトリ*</PDF_EXT_FONT>

保存ディレクトリ：外字ファイルの保存ディレクトリを絶対パスまたは相対パスで指定します。
セミコロン';'で区切る事により、複数ディレクトリを指定することができます。

4.4.1.4　Windowsシステム外字を使用する場合

- Windowsはデフォルト状態では外字がインストールされていないのでインストールする必要があります。またWindows外字サーバを用いる場合も同様です。

システム外字のフォント名は「EUDC」固定で、*Windowsシステムフォルダ*¥Fonts¥EUDC.TTEに保存されます。エクスプローラーには表示されませんが、コマンドプロンプトやWindows検索を実行すると表示することができます。

PDFを生成する場合は、EUDC.TTEをコピーして、拡張子を.ttfと変更し、EUDC.ttfの保存されているディレクトリを指定する必要があります。保存ディレクトリの指定については、『4.4.1.3　外字フォントの埋め込み』を参照してください。

コンフィグファイルの「fontNamesInfo.xml」に設定する必要はありません。

4.4.1.5　Windowsシステム外字以外を使用する場合

Windowsシステム外字以外の外字ファイルを使用することができます。
すでに他のアプリケーションでWindowsシステム外字を使用しており、XRF Readerでは別の外字ファイルを使用する必要がある場合に有効です。

手順は以下のとおりです。

手順1. 外字ファイルの保存

外字ファイルを以下のディレクトリに保存します。
　　インストールディレクトリ¥JRE¥LIB¥FONTS
外字ファイル名は任意につけることができますが、拡張子は「ttf」固定です。
拡張子「tte」では使用できないので、このような場合は「ttf」にリネームします。

手順2. フォント設定ファイルの設定

フォント設定ファイルで使用する外字フォントを指定します。

●フォント設定ファイル
フォント設定ファイルについては、『4.4　フォントについて』の『【フォント設定ファイル（fontNamesInfo.xml）】』を参照してください。

フォント設定ファイルの<GAIJI NAME=　/>タグで外字フォント名を指定します。たとえば外字フォント名が「mygaiji」の場合、以下のとおりに記述します。

●フォント設定ファイルの記述例

```
<!--　以下のTAGで、外字フォントを設定します　-->
<GAIJI NAME="mygaiji" ASCENT="880" DESCENT="144" />
```

これで、*インストールディレクトリ*¥JRE¥LIB¥FONTSに保存された外字フォントが使用されます。

JVMはシステムフォントよりも「*インストールディレクトリ*¥jre¥lib¥fonts」を優先的に参照します。外字ファイル名を「EUDC」として、「*インストールディレクトリ*¥jre¥lib¥fonts」にインストールした場合、XRFではシステム外字を参照できなくなります。

フォント名とファイル名には別の名前をつけることができます。
フォント名の指定は<GAIJI NAME= />タグで行いますが、ファイル名は任意につけることができます。

4.4.1.6　注意事項

外字コードの範囲はE000～F8FFのみとなります。

過去のベンダー定義文字にあるような未定義領域はサポートしていません。

SJISの外字領域（F040～F9FC）はUnicodeではE000～E757の領域に詰めて表現されます。

コード判定で英語・日本語以外のコードに対する考慮はされていません。

Unicodeの先頭のBOM（ef　bb　bf）は不要です。

4.4.2　外国語・多言語の扱いについて

ここでは、XRF Composerで外国語・多国語を含むファイルを生成する際に必要となる設定について説明します。

4.4.2.1　XRFとPDFにおける外国語・多言語の扱い

XRFファイルでは、ご使用のOSに多国語フォントがインストールされている必要があり、多国語フォントを埋め込むことはできません。

PDFファイルでは、多言語フォントを埋め込む必要があります。多言語フォントを埋め込まなくてもPDFは生成できますが、多言語フォントは表示されません。

4.4.2.2　外国語・多言語フォントの設定について

XRFで外国語フォントを利用する場合、外国語フォントをインストールして、*インストールディレクトリ*/conf/fontNamesInfo.xmlファイルにそのフォント情報を追記します。フォント設定ファイルについては、『4.4　フォントについて』の『【フォント設定ファイル（fontNamesInfo.xml）】』を参照してください。

また、多言語を同時に表現する場合は、多言語フォントをインストールしfontNamesInfo.xmlに設定を追加します。

4.4.2.3　多言語フォントの設定例

ここでは、多言語フォントである「Arial Unicode MS」を例にしてフォントの設定方法を説明します。多言語フォントを使用するためにはフォントファイルをインストールする必要があります。

例）Arial Unicode MSの追加

Arial Unicode MSを使用するためには、ARIALUNI.TTFが必要です。
ARIALUNI.TTFをインストールし、以下の設定をfontNamesInfo.xmlファイルに追記します。

```
<FONT NAME="Arial Unicode MS" ALIAS="Unicode" ASCENT="1491" DESCENT="430" />
```

※ 弊社では、日本語、英語、スペイン語、中国語繁体字、中国語簡体字、韓国語の動作を確認しています。

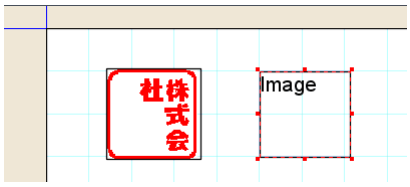
4.5　イメージデータについて

XRFで扱えるイメージデータの形式は以下のとおりです。

フォーム内にイメージデータを貼り付けることができます。
取り込むイメージには固定イメージとユーザイメージの2種類があります。

固定イメージは、フォーム設計時に貼り付けたイメージデータを画面上で確認できます。
しかし、固定イメージはリンク情報（イメージデータの保存場所）のみをフォームに取り込んでおり、イメージデータそのものは取り込まれていません。

一方ユーザイメージは、ユーザデータによって作成するためのもので、ユーザデータにより指定されたイメージファイルを貼り付けます。実際にユーザデータがマッピングされたときに内容が確定するため、フォーム設計段階では枠のみ表示されます。



- PNG形式（PNG）
- ビットマップ形式（BMP）
- JPEG形式（JPG）
- TIF形式（TIF, TIFF）
- GIF形式（GIF）

- イメージデータの作成手順については、『XRF Designer 操作マニュアル』の『6.　イメージの取り込み 』を参照してください。

4.6　バーコード

XRF Designerでは、フォーム内にバーコードを配置できます。
作成可能なバーコードオブジェクト形式と選択可能な種類は以下のとおりです。

＜形式＞

- 固定バーコード

  フォーム設計時にあらかじめバーコード化する値を設定します。
  フォーム作成時も同様の値でバーコード情報を生成します。

- ユーザバーコード

  フォーム設計時には、枠だけ表示されます。
  帳票生成時に、可変のユーザデータを入力してバーコードを形成するフィールドを作成します。

＜種類＞

NW-7	EAN8	EAN13	EAN128	ITF	IND25
Code39	Code128	QRコード	PDF417	郵政バーコード（カスタマバーコード）	

- バーコードの作成手順については、『XRF Designer 操作マニュアル』の『8.　バーコードの作成』を参照してください。

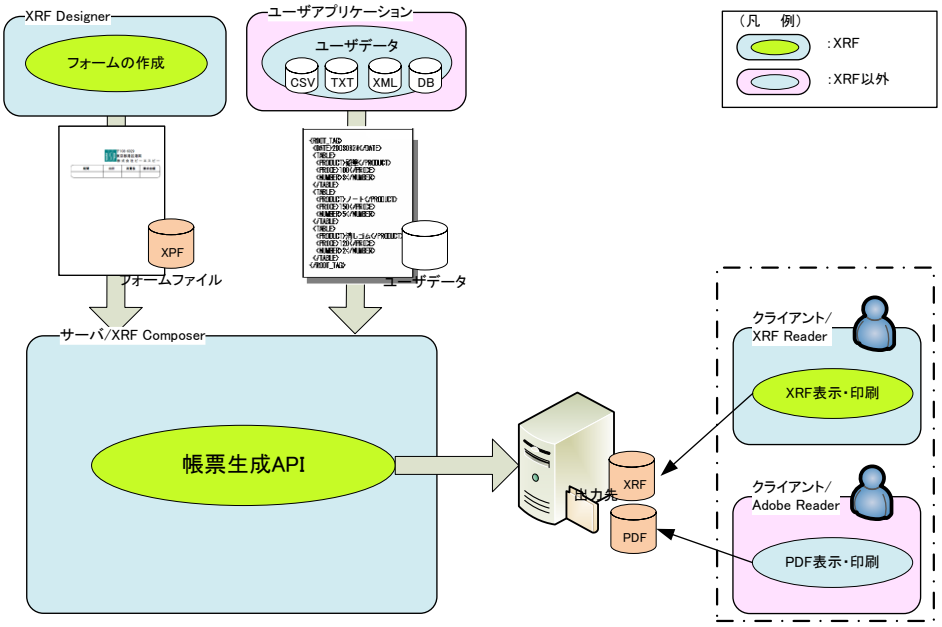
5.　帳票生成

ここでは、帳票生成の機能と動作について説明します。

5.1　帳票生成の機能

帳票生成は、XRF Designerで作成されたXPFフォームとお客様の持つユーザデータをXRF Composerによって重ね合わせXRF形式または、PDF形式の帳票を生成します。

帳票生成は、ジョブファイル生成やリモート処理と連携することもできます。

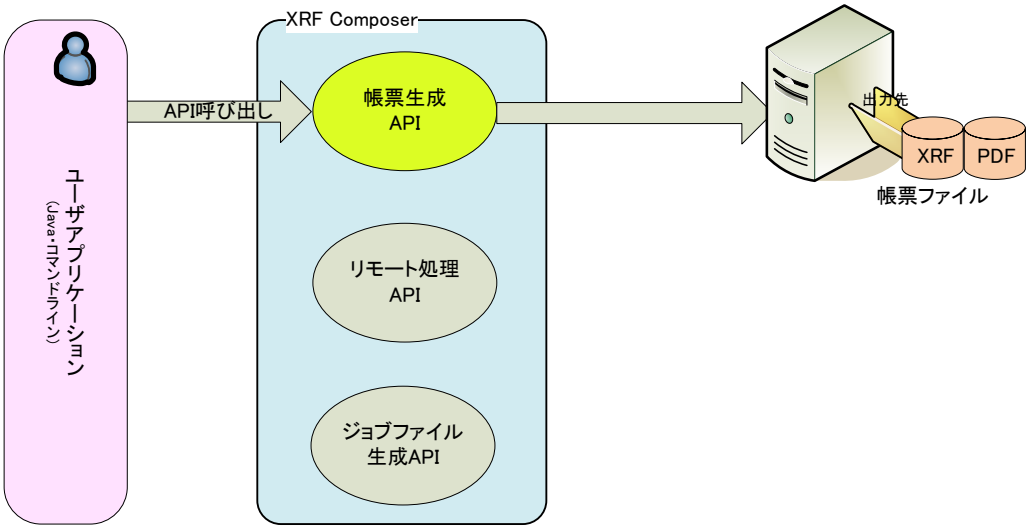


- 帳票生成の実行方法については、『XRF Composer 開発ガイド』の『2.2　帳票生成のAPI』を参照してください。

5.1.1 帳票生成の動作

帳票生成は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerは帳票生成APIに指定されているXPFファイルとユーザデータを重ね合わせXRF形式または、PDF形式の帳票を生成します。生成された帳票は、指定した出力先に保存されます。



5.2　ダイナミックオーバレイの機能

ここでは、ダイナミックオーバレイの機能と動作について説明します。

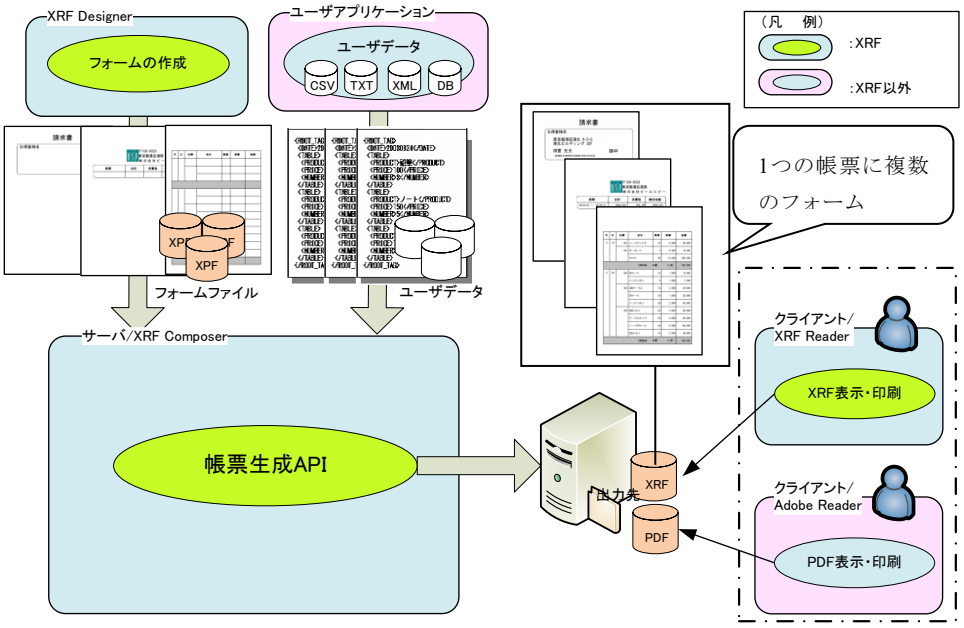
ダイナミックオーバレイは、複数のXPFフォームと対になるユーザデータをXRF Composerによって1つの帳票にまとめて生成する機能です。1つの帳票ファイルに複数のフォームを保有できます。たとえば、以下のような場合に活用できます。

例：送付用請求書の作成

1つ目のフォーム/送付先氏名、住所

2つ目のフォーム/支払日、請求金額

3つ目のフォーム/請求明細

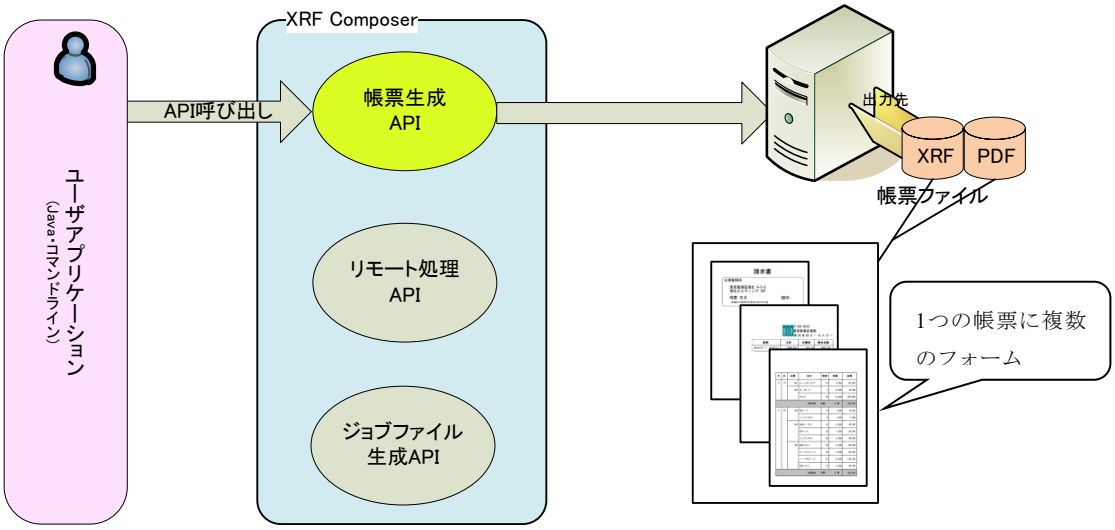


- ダイナミックオーバレイの実行方法については、『XRF Composer 開発ガイド』の『2.2.3 ダイナミックオーバレイによる帳票生成(複数入力)』と『2.2.4 ダイナミックオーバレイによる帳票生成(単一入力)』を参照してください。

5.2.1 ダイナミックオーバレイの動作

ダイナミックオーバレイは、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerは帳票生成APIに指定されている複数のXPFフォームと複数のユーザデータを重ね合わせ1つにまとめたXRF形式または、PDF形式の帳票を生成します。



6.　ジョブファイル

ジョブファイルは、複数のXRFファイルを格納し、それらの印刷要求を行うためのZIP形式のファイルです。

ここでは、ジョブファイルの機能と動作について説明します。

6.1　ジョブファイルの機能

ジョブファイルは、1つの処理要求で複数のXRFファイルの印刷が実行できます。

複数のXRFファイルまとめて格納することにより、各XRFファイルに印刷要求する必要はなく、１つの印刷要求で複数のXRFファイルを順番に印刷できます。また、ジョブファイルをコピーすることでバックアップファイルとしても活用できます。（バックアップは、手動で行います。）

このように、ジョブファイルは複数のXRFファイルを1つにまとめ、印刷順を守りたい場合などに便利な機能です。

ジョブファイルのファイル形式

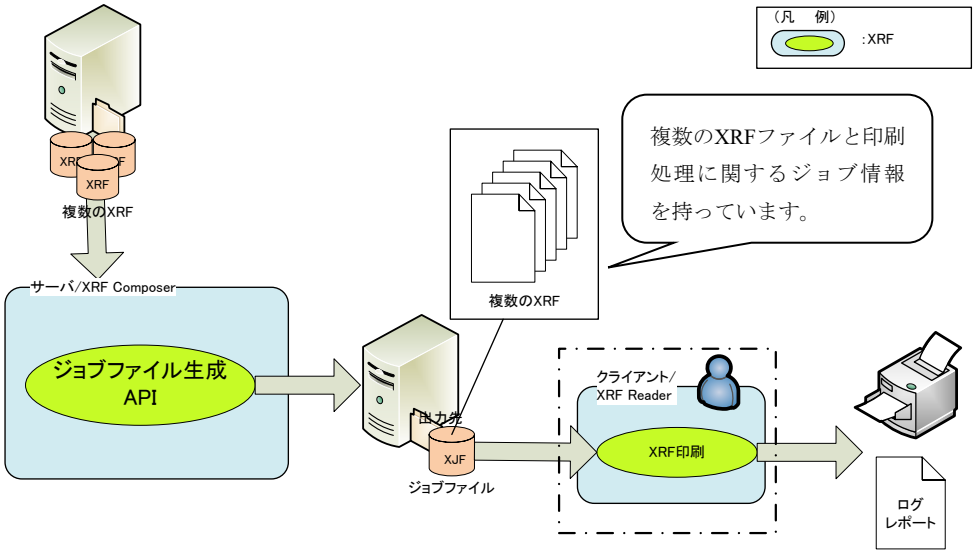
ジョブファイルは、拡張子が*.XJFのZIP形式のファイルです。

ジョブファイルの生成は、XRF Composerにより行われ、ジョブファイルの印刷は、XRF Readerによって行われます。

印刷完了後は、実行結果をジョブファイルごとにログレポートを出力します。

ログレポートの出力設定は、ジョブファイル生成時にXRF Composerで指定することができます。

帳票生成やリモート処理と連携することもできます。

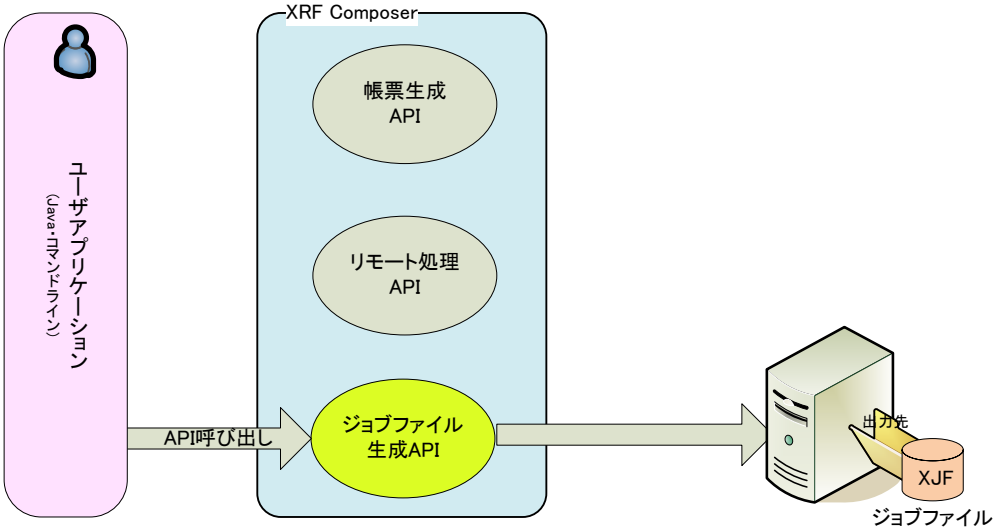


- ジョブファイル生成の実行方法やログレポートの出力設定については、『XRF Composer 開発ガイド』の『4.2　ジョブファイル生成パラメータ』を参照してください。
- ジョブファイルの印刷は、リモート処理で実行します。

6.1.1 ジョブファイル生成の動作

ジョブファイル生成は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerはジョブファイル生成APIに指定されている複数のXRFファイルをジョブファイル生成します。ジョブファイルは、指定した出力先に保存されます。



7.　リモート処理（XRF Remote Launch Broker）

リモート処理（XRF Remote Launch Broker）は、XRFの分散印刷（リモート印刷）とリモート画面表示を提供できるサービスです。

ここでは、リモート処理の機能と動作について説明します。

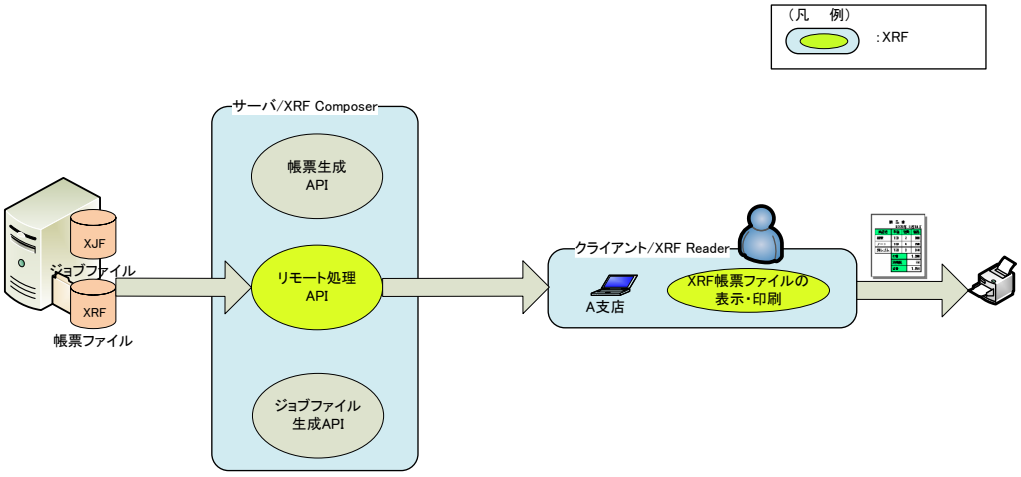
7.1　リモート処理（XRF Remote Launch Broker）の機能

リモート処理（XRF Remote Launch Broker）は、XRFの分散印刷（リモート印刷）とリモート画面表示機能です。

XRF Composerが動作するマシンと同じLAN/WAN内のマシンのXRF Readerに対して印刷を実行します。また、LAN/WAN内にWebサーバを配置し、XRF ReaderにURLを指定して印刷することもできます。

リモート処理は、帳票生成やジョブファイル生成と連携することもできます。

リモート処理（XRF Remote Launch Broker）を実行する際は、XRF Readerを常駐している必要があります。常駐の設定方法については、『XRF Reader操作マニュアル』の『8.　XRF Readerの常駐機能』を参照してください。

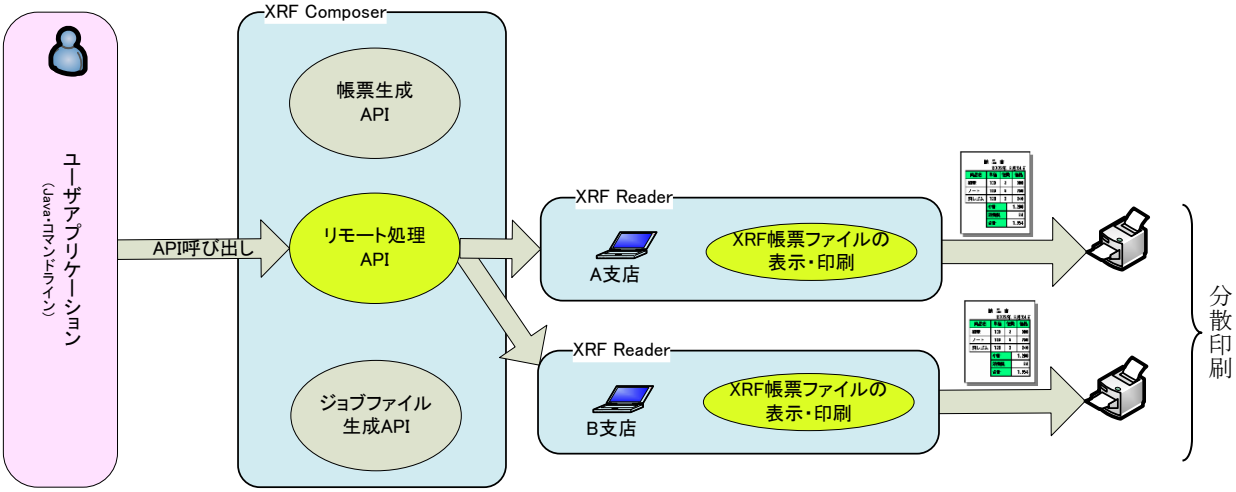


リモート処理の実行方法については、『XRF Composer 開発ガイド』の『5.　リモート処理（XRF Remote Launch Broker）のAPI』を参照してください。

7.1.1　リモート処理で帳票ファイルを分散印刷の動作

帳票ファイルを分散印刷する場合は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerはリモート処理APIに指定されたXRF Readerに対して帳票ファイル（XRF）を印刷要求します。指定された帳票ファイルは、XRF Readerから印刷します。

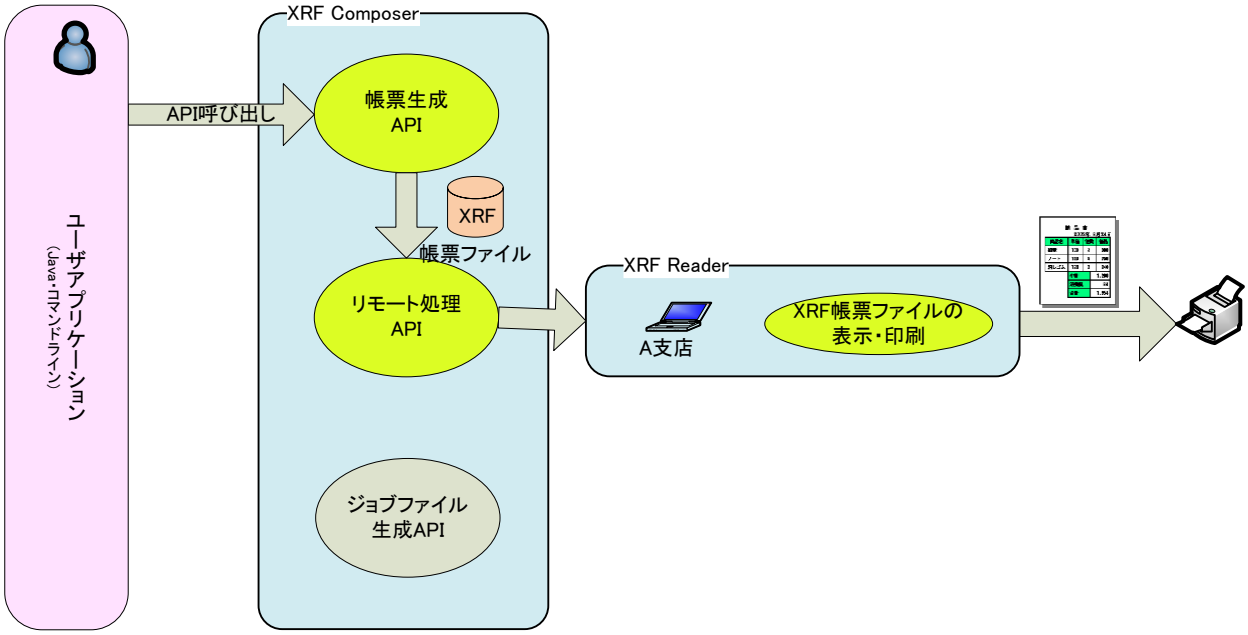


7.1.2　リモート処理で帳票生成から印刷までの動作

帳票生成とリモート処理を連携することで、XRF形式の帳票生成から印刷まで実行できます。

帳票生成とリモート処理の連携は、以下のとおりに動作します。

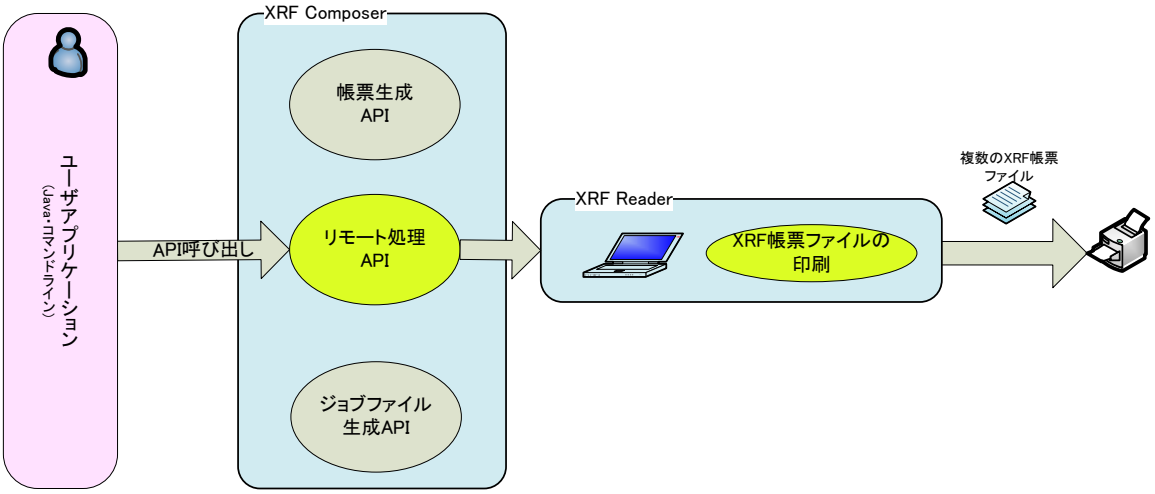
ユーザアプリケーションからAPIを呼び出すと、XRF Composerは帳票生成APIに従いXRF形式の帳票を生成します。生成された帳票は、リモート処理APIに指定されたXRF Readerに対して印刷要求します。XRF Readerから帳票を印刷します。



7.1.3　リモート処理でジョブファイルを印刷の動作

ジョブファイルを印刷する場合は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerはリモート処理APIに指定されたXRF Readerに対してジョブファイルを印刷要求します。XRF Readerは、ジョブファイルに格納されたXRFファイルを順番に印刷します。

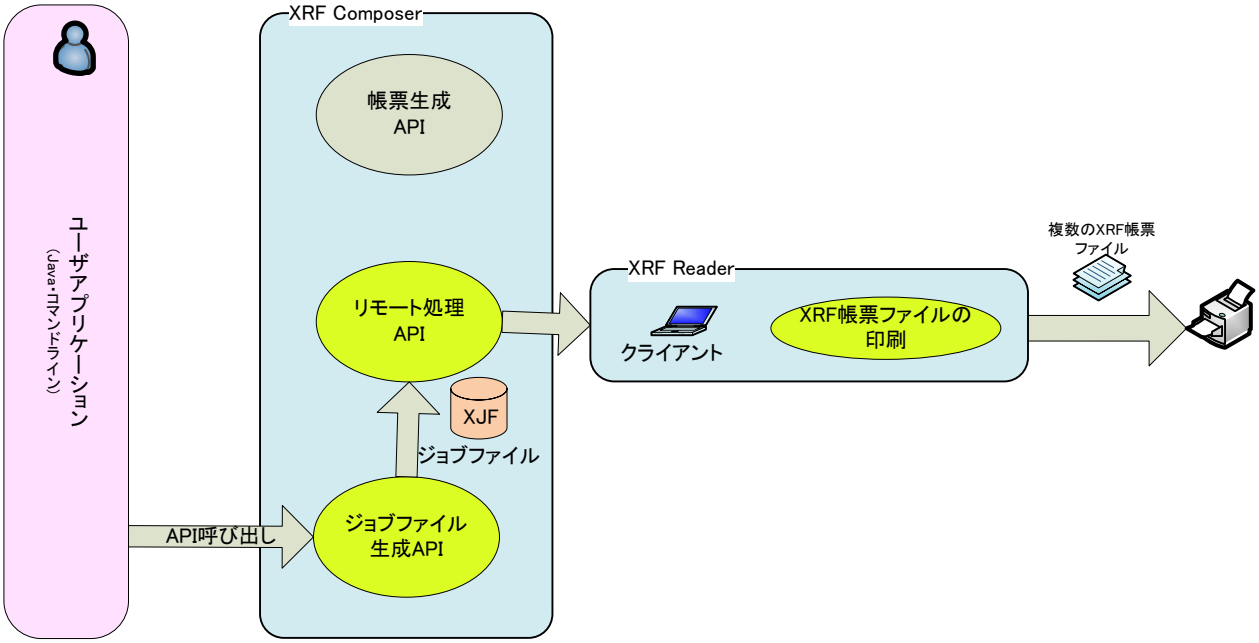


7.1.4　リモート処理でジョブファイル生成して印刷の動作

ジョブファイル生成とリモート処理を連携することで、ジョブファイル生成から印刷まで実行できます。

ジョブファイルとリモート処理の連携は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerはジョブファイル生成APIに指定されている複数のXRFファイルをジョブファイル生成し、リモート処理APIに指定されたXRF Readerに対してジョブファイルを印刷要求します。XRF Readerは、ジョブファイルに格納されたXRFファイルを順番に印刷します。



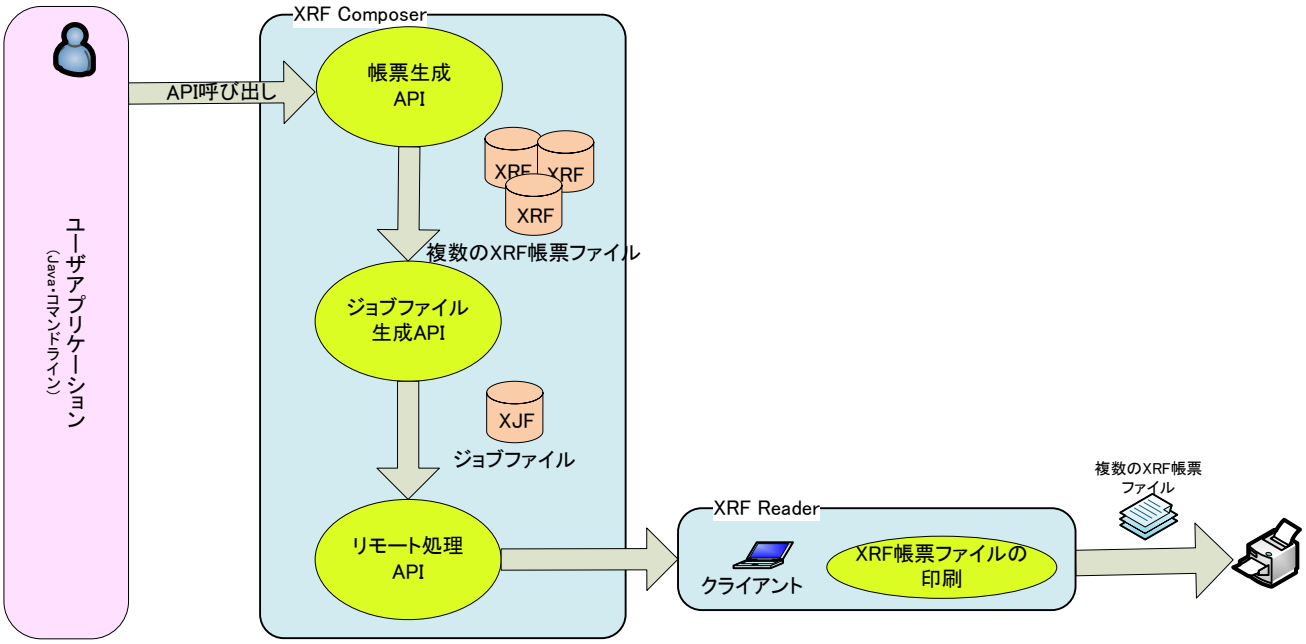
- ジョブファイル生成とリモート処理を連携した場合、印刷完了後にジョブファイルは削除されます。ジョブファイルを保存したい場合は、ジョブファイル生成とリモート処理を別々に行ってください。ジョブファイル生成については、『6.　ジョブファイル』を参照してください。

7.1.5　リモート処理で帳票生成、ジョブファイル生成して印刷実行の動作

帳票生成、ジョブファイル生成、リモート処理3つの機能を連携することで、XRF形式の帳票生成、ジョブファイル生成、印刷までをまとめて実行できます。

帳票生成、ジョブファイル生成、およびリモート処理の連携は、以下のとおりに動作します。

ユーザアプリケーションからAPIを呼び出すと、XRF Composerは帳票生成APIに指定されている複数のXPFフォームとユーザデータを重ね合わせXRF形式の帳票を生成します。生成された帳票は、ジョブファイル生成APIに指定されている複数のXRF帳票ファイルをジョブファイル生成し、リモート処理APIに指定されたXRF Readerに対してジョブファイルを印刷要求します。XRF Readerは、ジョブファイルに格納されたXRFファイルを順番に印刷します。



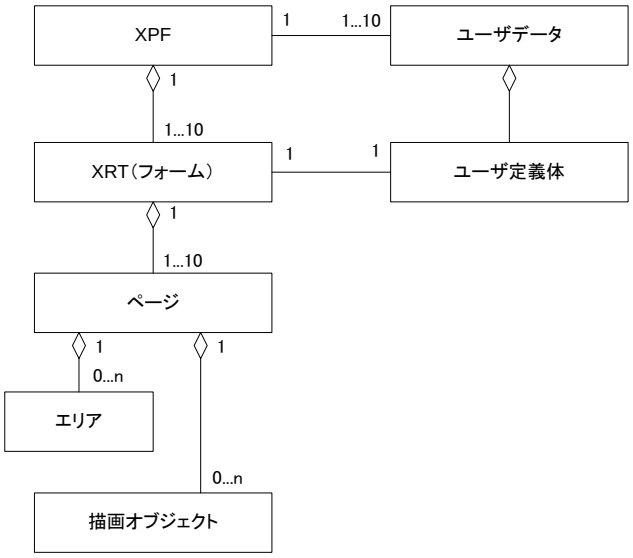
- 帳票生成、ジョブファイル生成、リモート処理3つの機能を連携した場合、印刷完了後にジョブファイルは削除されます。ジョブファイルを保存したい場合は、ジョブファイル生成とリモート処理を別々に行ってください。
  ジョブファイル生成については、『6.　ジョブファイル』を参照してください。

8.　論理データ

ここではXRF Designerで生成されるXRTに関係する論理データについて説明します。

8.1　データ構造と各データの説明

XRTに関係する論理データの構造を以下に示します。



XRF

XRFはXRF Composerにより生成され、XRT（フォーム）・ユーザデータ・イメージデータ・データ定義体などを保持する帳票データファイルです。

XRFには複数のXRT（フォーム）・ユーザデータ（XML、CSV、TEXT）を保持することができます。

XRT

XRF Designerで保存したフォームファイルの形式です。

XRTは1つのフォームのみを保持します。

フォーム

フォームにはシンプルフォームとトレースフォームがあります。

シンプルフォームにはサイクリックオプションを指定することができます。

複数のページを保持することができます。

データ定義体

データ定義体はユーザデータの構造を規定するものです。

データ定義体とフォームは1対1で対応します。

フォームの構造やテーブルの有無によって、データ定義体の構造が制限されます。

ユーザデータ

ユーザアプリケーションで作成し、データフィールドに対応付けるデータです。

XML形式、CSV形式、TEXT形式で記述することができます。また、クエリによるDBアクセスも可能です。

ユーザデータはXRTに対して指定します。

ページ

描画オブジェクト等を配置できる、描画のベースとなるものです。

ベースエリアを含む1つ以上のエリアと0個以上の描画オブジェクトを保持します。

エリア

ページとは独立して、描画オブジェクトを配置できる領域です。

エリアのみを部品として別に保存することもできます（共有エリア）。

「共有エリア」をリンク参照することもできます。

（ただしデータフィールドをもつ共有エリアをリンクで利用することはできません）

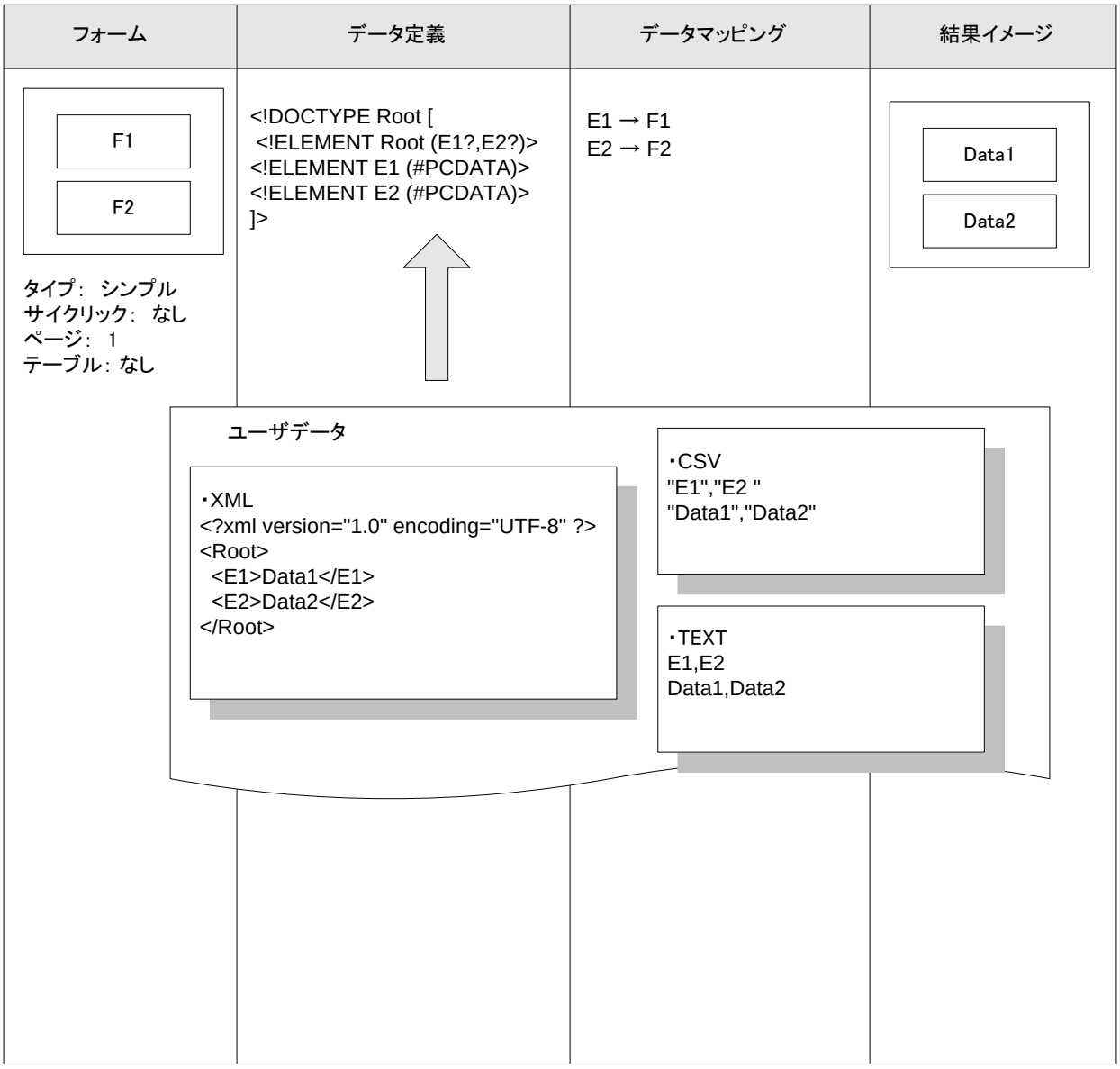
「テーブル」はエリアの一種として扱います。

描画オブジェクト

図形、データフィールドなど描画するためのオブジェクトです

XRFで扱えるオブジェクトは、以下のとおりです。
弧（弓、扇も含む）、曲線、円、直線、平行線、連続線、多角形、四角形、固定イメージ、固定テキスト、ユーザフィールド（文字列、数値、日付）、ユーザイメージ、マスク、ページ番号、グラフ、固定バーコード、ユーザバーコード

8.1.1　パターン例1 -- データフィールドだけがある場合



8.1.2　パターン例2 -- 固定テーブルが2つある場合

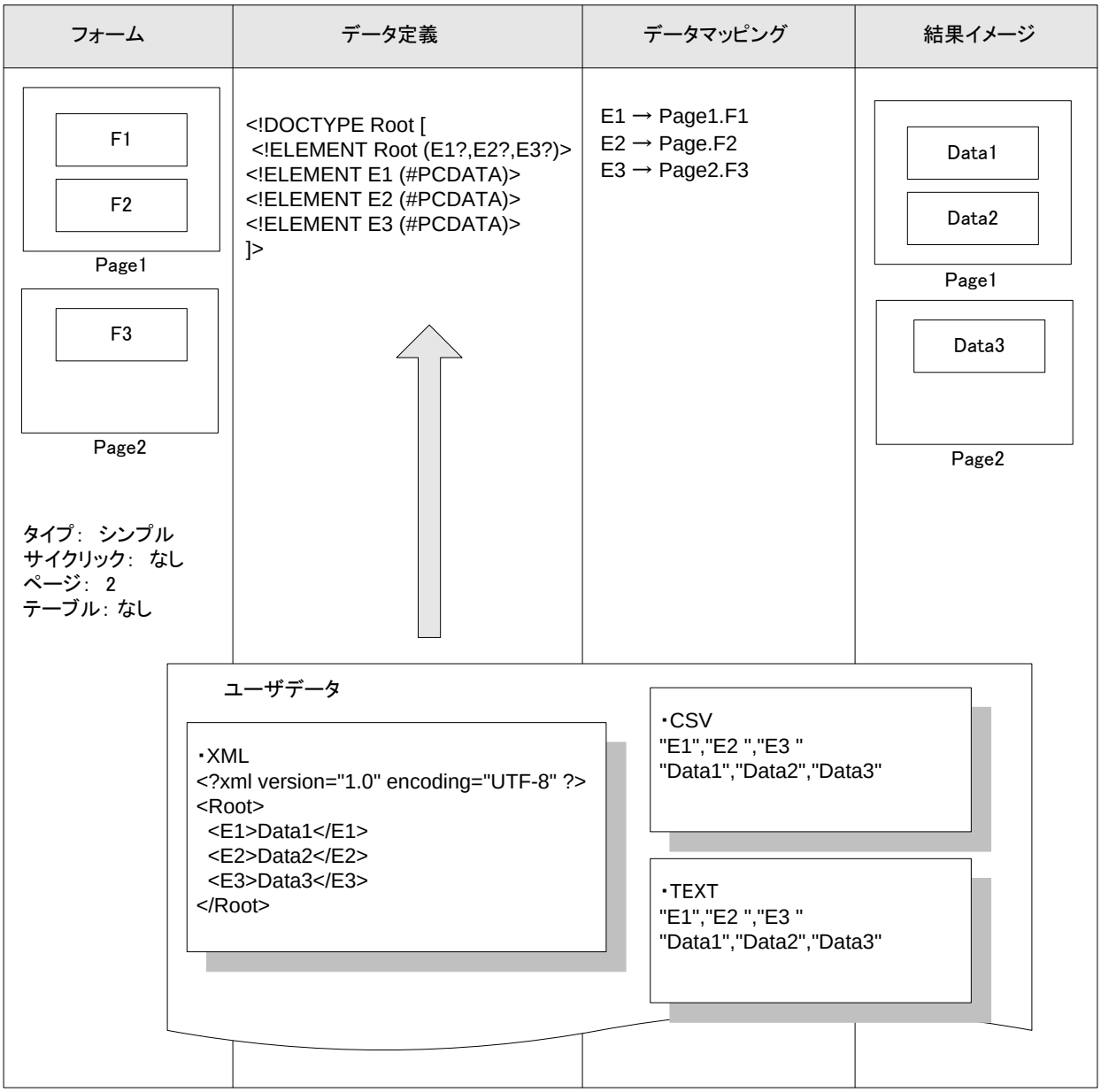
フォーム	データ定義	データマッピング	結果イメージ
Table1 Table2 タイプ： シンプル サイクリック： なし ページ： 1 テーブル：固定2	<!DOCTYPE Root [<!ELEMENT Root (record1* , record2*)> <!ELEMENT record1 (E1?,E2?)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)> <!ELEMENT record2 (E3?,E4?)> <!ELEMENT E3 (#PCDATA)> <!ELEMENT E4 (#PCDATA)>]>	record1 → Table1 record2 → Table2 E1→Table1.F1 E2→Table1.F2 E3→Table2.F3 E4→Table2.F4	Table1 Data11 Data21 Data12 Data22 Table2 Data31 Data41 Data32 Data42

ユーザデータ
※CSVは不可

・XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<Root>
  <record1>
    <E1>Data11</E1> <E2>Data21</E2>
  </record1>
  <record1>
    <E1>Data12</E1> <E2>Data22</E2>
  </record1>
  <record2>
    <E3>Data31</E3><E4>Data41</E4>
  </record2>
  <record2>
     <E3>Data32</E3><E4>Data42</E4>
  </record2>
</Root>
```

8.1.3　パターン例3 -- 2ページの場合



8.1.4　パターン例4 -- 可変テーブルの場合

フォーム	データ定義	データマッピング	結果イメージ
Table1 Page1 Table2 Page2 タイプ： シンプル サイクリック： なし ページ： 2 テーブル：可変2	<!DOCTYPE Root [<!ELEMENT Root (E1?,E2?)> <!ELEMENT E1 (#PCDATA)> <!ELEMENT E2 (#PCDATA)>]>	record1 → Table1 E1 → Page1.Table1.F1 E2 → Page1.Table1.F2	Table1 D11 D21 D12 D22 Table2 D13 D23 D14 D24 D15 D25 D16 D26 Table2 D17 D27 D18 D28

ユーザデータ

・XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<Root>
 <record1>
  <E1>D11</E1> <E2>D21</E2>
 </record1>
 <record1>
  <E1>D12</E1> <E2>D22</E2>
 </record1>
 <record1>
  <E1>D13</E1> <E2>D23</E2>
 </record1>
 <record1>
  <E1>D14</E1> <E2>D24</E2>
 </record1>
 <record1>
  <E1>D15</E1> <E2>D25</E2>
 </record1>
 <record1>
  <E1>D16</E1> <E2>D26</E2>
 </record1>
 <record1>
  <E1>D17</E1> <E2>D27</E2>
 </record1>
 <record1>
  <E1>D18</E1> <E2>D28</E2>
 </record1>
</Root>
```

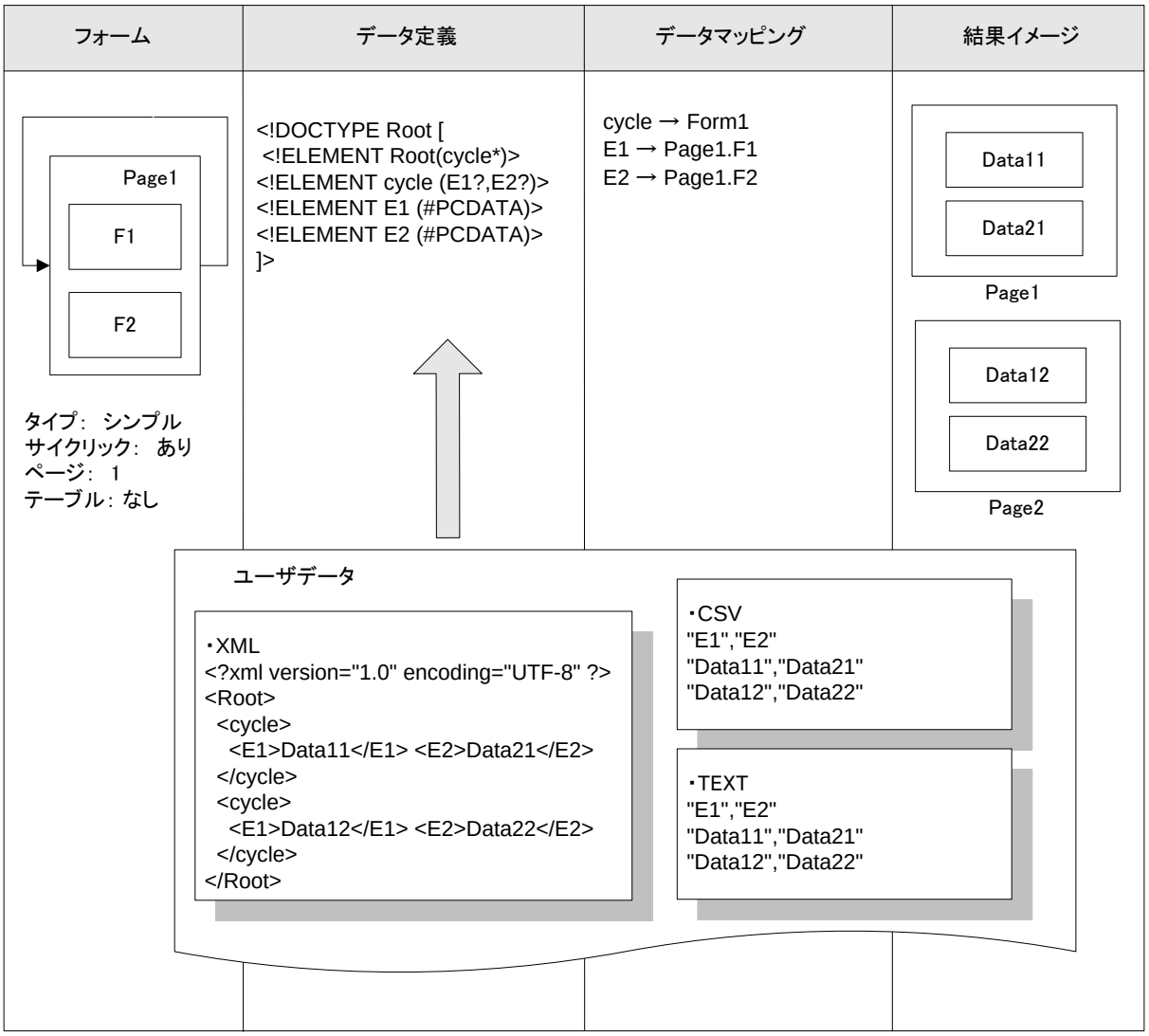
・CSV

```
"E1","E2"
"D11","D21"
"D12","D22"
"D13","D23"
"D14","D24"
"D15","D25"
"D16","D26"
"D17","D27"
"D18","D28"
```

・TEXT

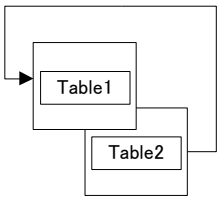
```
"E1","E2"
"D11","D21"
"D12","D22"
"D13","D23"
"D14","D24"
"D15","D25"
"D16","D26"
"D17","D27"
"D18","D28"
```

8.1.5　パターン例5 -- 1ページをサイクリックする場合



8.1.6　パターン例6 -- 2ページをサイクリックする場合

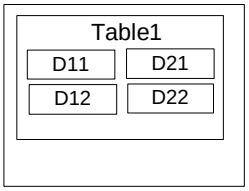
フォーム	データ定義	データマッピング	結果イメージ



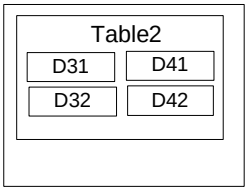
タイプ： シンプル
サイクリック： あり
ページ： 2
テーブル：固定2

```
<!DOCTYPE Root [
<!ELEMENT Root (cycle*)>
<!ELEMENT cycle
 (record1*,record2*)>
<!ELEMENT record1 (E1?,E2?)>
<!ELEMENT E1 (#PCDATA)>
<!ELEMENT E2 (#PCDATA)>
<!ELEMENT record2 (E3?,E4?)>
<!ELEMENT E3 (#PCDATA)>
<!ELEMENT E4 (#PCDATA)>
]>
```

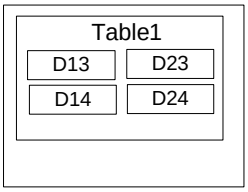
cycle → Form1
record1→Table1
E1→Page1.Table1.F1
E2→Page1.Table1.F2
record2→Table2
E3→Page2.Table2.F3
E4→Page2.Table2.F4



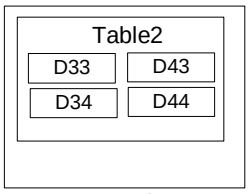
page1



page2



page3



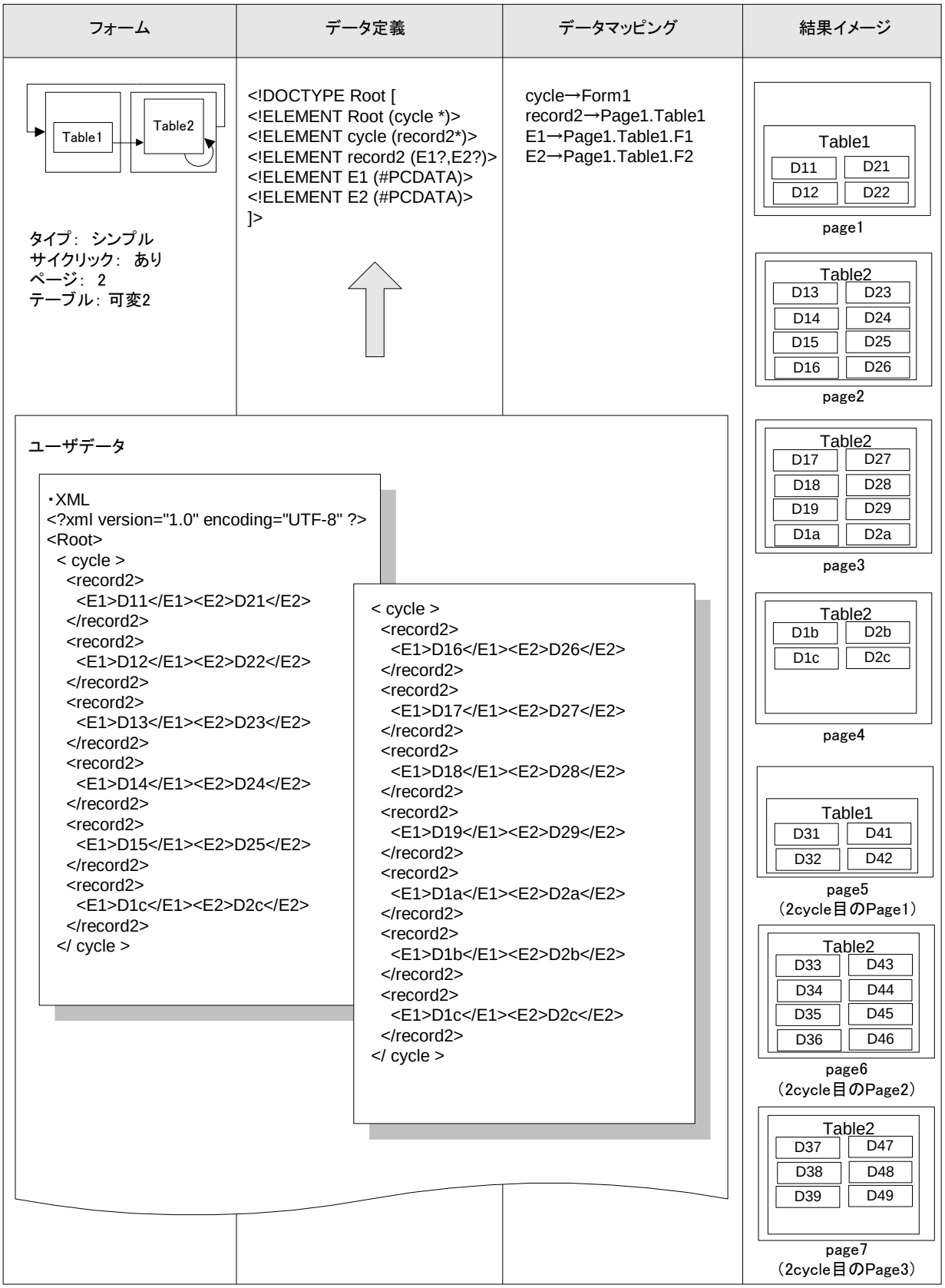
page4

ユーザデータ

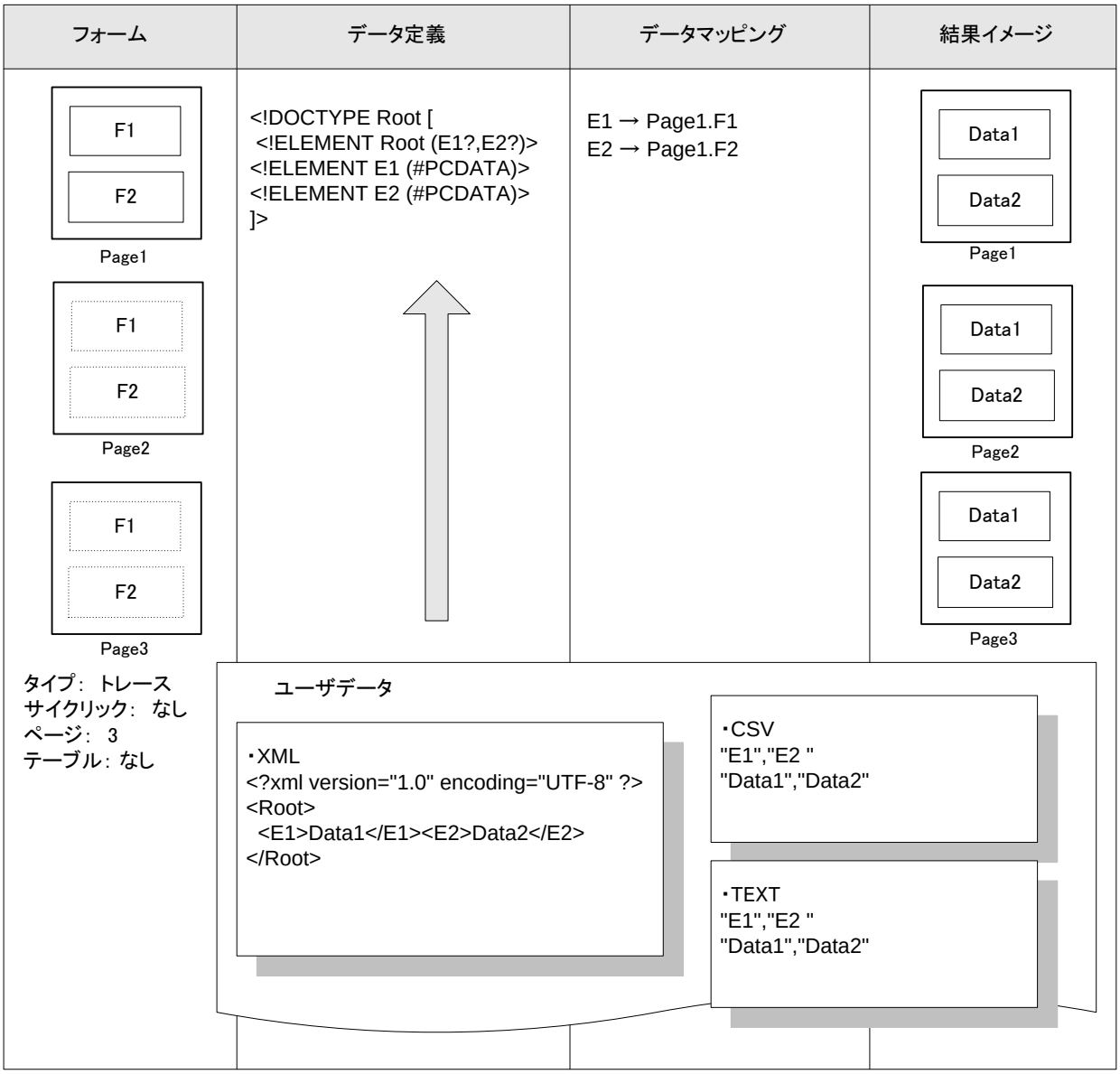
・XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<Root>
 < cycle >
  <record1>
   <E1>D11</E1><E2>D21</E2>
  </record1>
  <record1>
   <E1>D12</E1><E2>D22</E2>
  </record1>
  <record2>
   <E3>D31</E3><E4>D41</E4>
  </record2>
  <record2>
   <E3>D32</E3><E4>D42</E4>
  </record2>
 </ cycle >
 <cycle >
  <record1>
   <E1>D13</E1><E2>D23</E2>
  </record1>
  <record1>
   <E1>D14</E1><E2>D24</E2>
  </record1>
  <record2>
   <E3>D33</E3><E4>D43</E4>
  </record2>
  <record2>
   <E3>D34</E3><E4>D44</E4>
  </record2>
 </ cycle >
</Root>
```

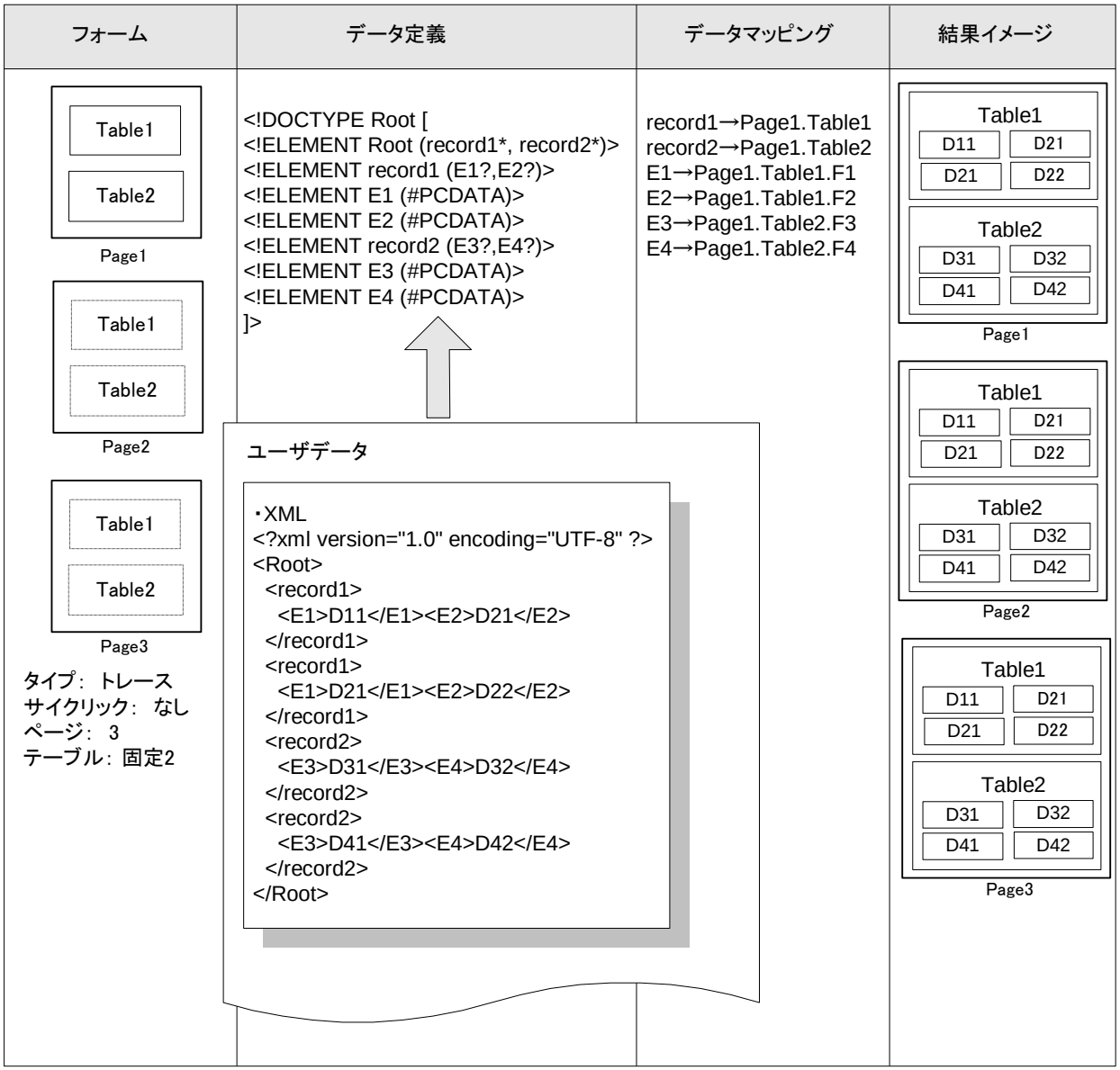
8.1.7　パターン例7 -- 可変テーブルがあるものをサイクリックする場合



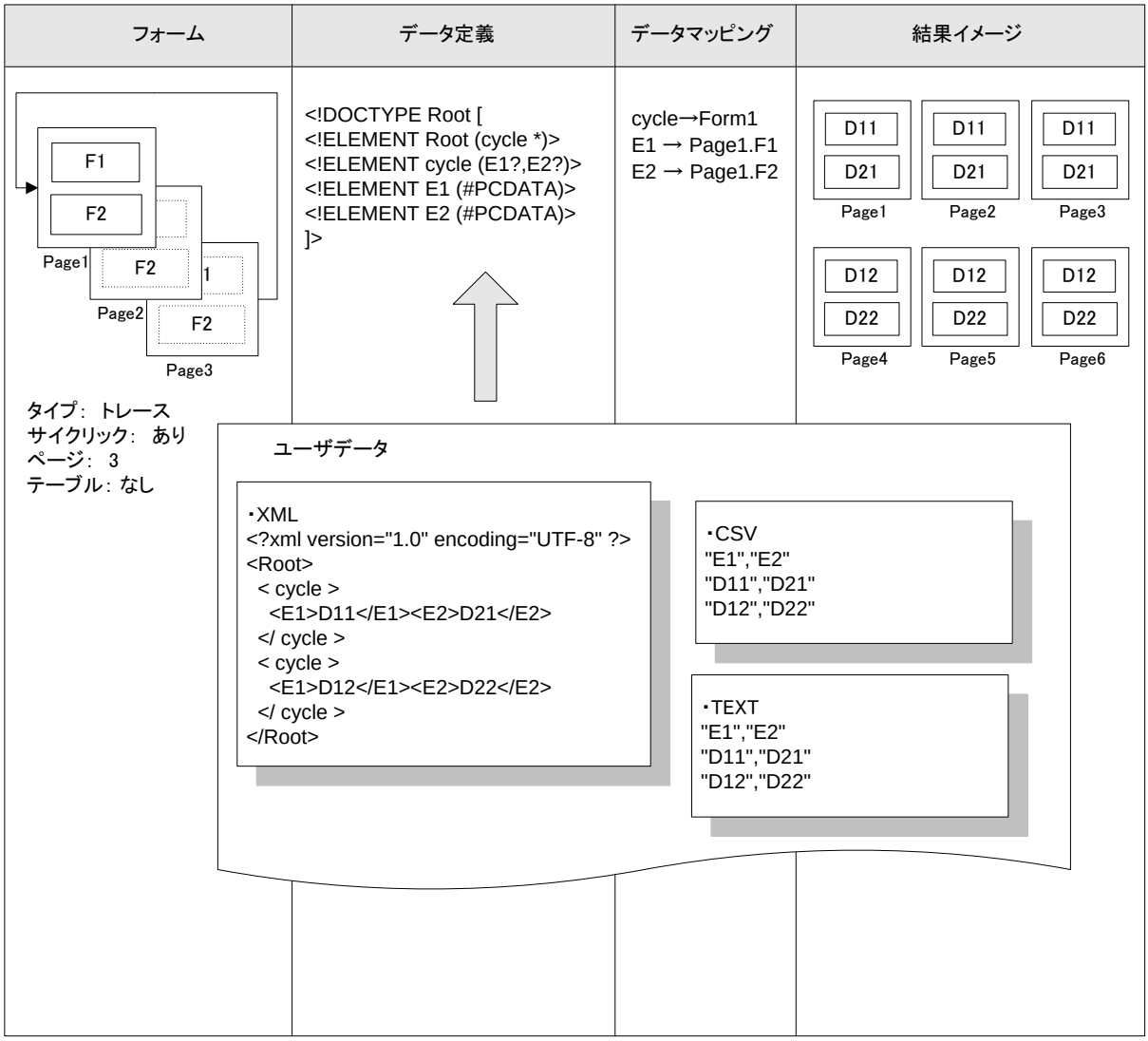
8.1.8　パターン例8 -- 複写帳票でフィールドしかない場合



8.1.9　パターン例9 -- 複写帳票でテーブルがある場合

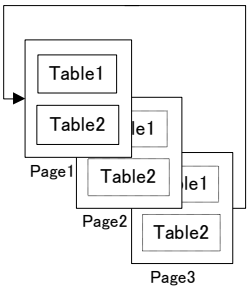


8.1.10　パターン例10 -- 複写帳票をサイクリックする場合



8.1.11　パターン例11 -- テーブルがある複写帳票をサイクリックする場合

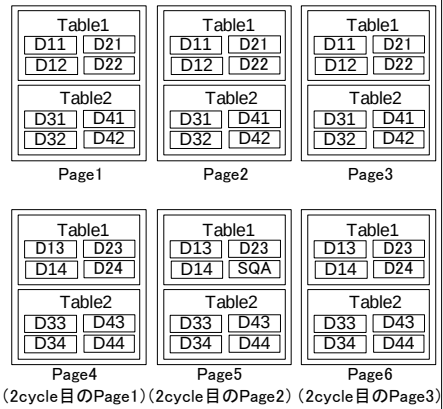
フォーム	データ定義	データマッピング	結果イメージ



タイプ：トレース
サイクリック：あり
ページ：3
テーブル：固定2

```
<!DOCTYPE Root [
<!ELEMENT Root (cycle *)>
<!ELEMENT cycle (record2*,
record3*)>
<!ELEMENT record2
(E1?,E2?)>
<!ELEMENT E1 (#PCDATA)>
<!ELEMENT E2 (#PCDATA)>
<!ELEMENT record3
(E3?,E4?)>
<!ELEMENT E3 (#PCDATA)>
<!ELEMENT E4 (#PCDATA)>
]>
```

cycle→Form1
record2→Page1.Table1
record3→Page1.Table2
E1→Page1.Table1.F1
E2→Page1.Table1.F2
E3→Page1.Table2.F3
E4→Page1.Table2.F4



ユーザデータ

・XML

```
<?xml version="1.0" encoding="UTF-8"?>
<Root>
 < cycle >
  <record2>
   <E1>D11</E1><E2>D21</E2>
  </record2>
  <record2>
   <E1>D12</E1><E2>D22</E2>
  </record2>
  <record3>
   <E3>D31</E1><E2>D41</E2>
  </record3>
  <record3>
   <E3>D32</E1><E2>D42</E2>
  </record3>
 </ cycle >
 < cycle >
  <record2>
   <E1>D13</E1><E2>D23</E2>
  </record2>
  <record2>
   <E1>D14</E1><E2>D24</E2>
  </record2>
  <record3>
   <E3>D33</E1><E2>D43</E2>
  </record3>
  <record3>
   <E3>D34</E1><E2>D44</E2>
  </record3>
 </ cycle >
</Root>
```

索 引

C

CSVデータ 19

X

XMLデータ 17
XRF 46
XRF Reader 1
XRF Composer 3
XRF Designer 3
XRFの運用構成例 4
XRFの概要 1
XRFの構成 2
XRFの構成要素 3
XRFの最小構成 4
XRFのシステム構成図 2
XRFの大規模構成 6
XRFの中規模構成 5
XRFの特徴 1
XRF Viewer 3
XRF Reader 3
XRT 46

い

イメージ 34

え

エリア 47

か

外国語・多言語の扱い 33
外字 30

さ

サイクリック形式 9
サポートされるデータ定義体の形式 16

し

ジョブファイル 39
ジョブファイル生成の動作 40
ジョブファイルの機能 39
シンプルフォーム 7

た

対応可能な帳票形式 11
対応可能な帳票の種類 7
対応可能なデータの種類 15
ダイナミックオーバレイの機能 37
ダイナミックオーバレイの動作 38

ち

帳票生成 35
帳票生成の機能 35
帳票生成の動作 36

て

データ定義体 15, 46
TEXTデータ 23

と

トレースフォーム 8

は

バーコード 34

ひ

描画オブジェクト 47

ふ

フォーム 46
フォームの種類と形式 7
フォント 30

へ

ページ 47

ゆ

ユーザデータ 17, 46

ら

ラインデータフォーム 10

り

リモート処理（XRF Remote Launch Broker） 41
リモート処理（XRF Remote Launch Broker）の機能 41
リモート処理でジョブファイル生成して印刷の動作 .. 44
リモート処理でジョブファイルを印刷の動作 43
リモート処理で帳票生成、ジョブファイル生成して印刷実行の動作 45
リモート処理で帳票生成から印刷までの動作 43
リモート処理で帳票ファイルを分散印刷の動作 42

ろ

ロット化形式 9

XRF V3.8 ユーザーズマニュアル

2022年9月　第2版

発行：株式会社ユニリタ

〒108-6029　東京都港区港南2-15-1　品川インターシティA棟

Copyright © 2022 株式会社ユニリタ

本書は著作権上の保護を受けています。本書の一部あるいは全部について、株式会社ユニリタから文書による許諾を受けずに、いかなる方法においても複写・複製・転載することは禁じられています。

改良のため、製品の仕様やデザインは予告なしに変更される場合がありますのでご了承ください。