

**Micro Focus Visual COBOL 2.1J**  
**SAP Crystal Reports for Visual Studio 2010**  
**動作検証 検証結果報告書**

**2013 年 5 月 1 日**

**マイクロフォーカス株式会社**

Copyright © 2013 Micro Focus. All Rights Reserved.

記載の会社名、製品名は、各社の商標または登録商標です。

## 1. 検証概要、目的及びテスト方法

### 1.1 検証概要

SAP Crystal Reports は全世界で大きなシェアを持つレポートツールであり、多種多様なデータソースを多角的な観点から容易にビジュアル化します。本稿執筆時点で最新版の 13 では flash との統合機能、WPF Viewer、ソートコントロールなど最先端で且つ高機能なレポート機能を実装しています。

一方従来 COBOL 言語で書かれてきた多くのバッチ処理も、帳票出力という形で企業情報をビジュアル化する処理を行ってきました。このような COBOL で書かれたビジネスロジックは企業活動そのものを体現するものとして企業の大切な経営資源となっています。

企業のコアとなる COBOL ビジネスロジックと Crystal Reports の先端的なビジュアル化機能を組み合わせることによって、モダナイズされた戦略的情報システムをこれまでの IT 投資により積み上げられてきたアセットを活用しつつ構築することができます。

ここでは、SAP Crystal Reports を使用した Web レポートのアプリケーションで COBOL のビジネスロジックを活用する方法を検証しました。

### 1.2 目的及びテスト方法

SAP Crystal Reports は Visual Studio IDE や Eclipse IDE との統合機能を提供しています。本報告書では Visual Studio IDE に組み込んで使用する SAP Crystal Report, version for Visual Studio .NET と連携した .NET ベースの Web レポートのアプリケーションで COBOL のビジネスロジックを活用できることを検証します。

## 2. 検証環境

### ソフトウェア

Windows 7 Professional x86 (32-Bit)

Microsoft Visual Studio 2010 Professional, Version 10.0.30319

Microsoft SQL Server 2008 Express

SAP Crystal Reports for Visual Studio 2010, Version 13.0.0.99

Micro Focus Visual COBOL 2.1J

### ハードウェア

機種 : Dell OPTIPLEX7010

CPU : Intel Core2 i7-3770 3.40GHz

Memory : 2.00 Gbyte memory

### 3. テスト内容

COBOL データファイルに格納されたデータを基に SAP Crystal Report 帳票が生成できることを検証用のアプリケーションを使用して検証します。本検証用アプリケーションにおける Crystal Reports オブジェクトは COBOL データファイルより読み込んだデータがセットされた .NET のデータテーブルオブジェクト参照して帳票を生成しています。

### 4. 結果

#### 4.1 検証環境の準備

SQL Server 2008 のサンプルデータベース AdventureWorksLT2008 をダウンロードして SQL Server 2008 Express に配備しました。

ダウンロード元(2013/5/1 リンク検証) :

<http://msftdbprodsamples.codeplex.com/downloads/get/106391>

#### 4.2 インストール

SAP Crystal Reports for Visual Studio 2010, Version 13.0.0.99 をダウンロードしてインストールしました。

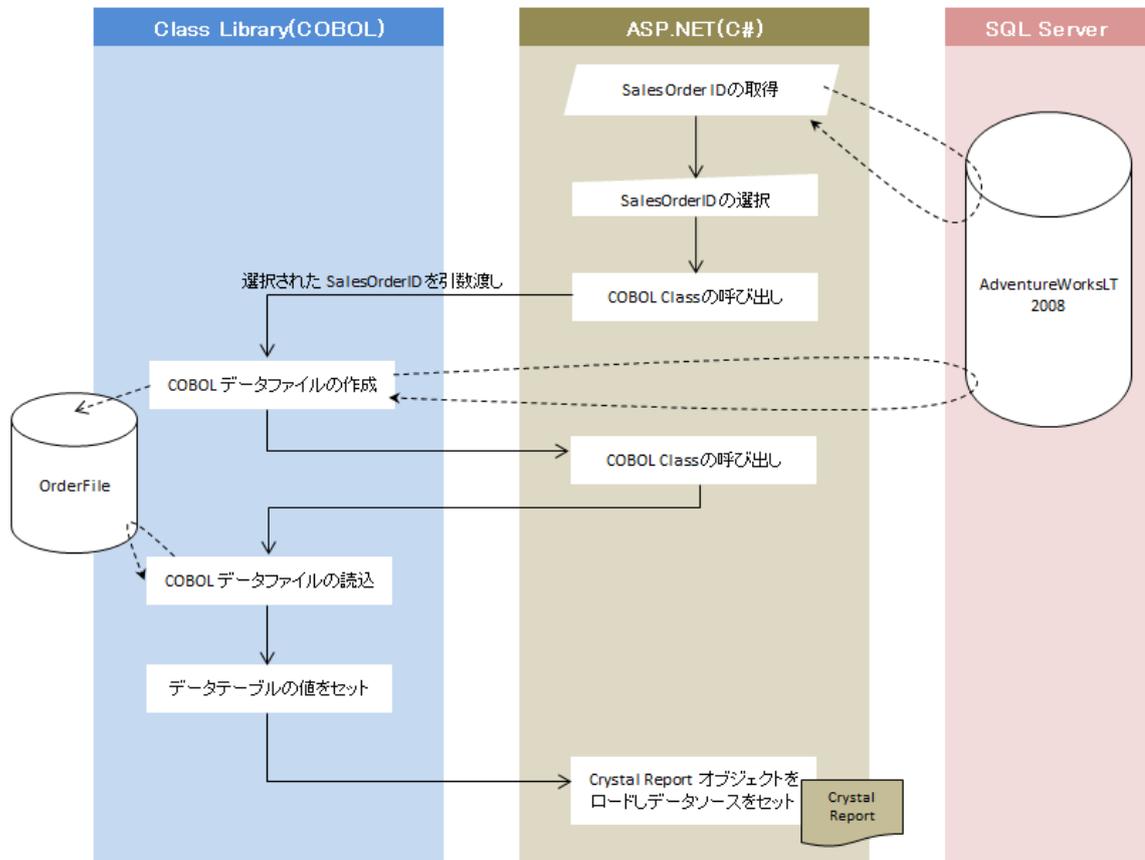
ダウンロード元(2013/5/1 リンク検証) :

[http://downloads.businessobjects.com/akdlm/cr4vs2010/CRforVS\\_13\\_0.exe](http://downloads.businessobjects.com/akdlm/cr4vs2010/CRforVS_13_0.exe)

#### 4.3 サンプルアプリケーションの作成

本検証では下図のようなフローで Crystal Reports を動的に生成する Web Application を作成して動作を確認しました。

図 4-1 サンプルアプリケーションの処理フロー



アプリケーションが使用する主なプログラムは下表の通りとなります。

表 4-1 プログラム一覧

プログラム名	主な機能
CobFileHandle.cbl	CreateCobFile メソッド、ReadCobFile メソッドを含む COBOL で書かれたマネージド Class。
OrderFILE.dat	COBOL プログラムにより処理されるレコード順ファイル。
Default.aspx	ASP.NET のプレゼンテーションファイル
Default.aspx.cs	ASP.NET のコードビハインド。SalesOrderID の一覧を表示するドロップダウンのイベントハンドラ lstChanged が COBOL Class の呼び出しや Crystal Reports のロード等処理。
CR_By_Order_ID.rpt	Crystal Reports のエンティティ。

DS\_By\_Order\_ID.xsd

CR\_By\_Order\_ID.rpt に紐つけられた表示用のデータセット。

---

実際に使用したアプリケーションは、Micro Focus のウェブ上に本報告書と共に公開しています。

#### 4.4 サンプルアプリケーションの実行結果

サンプルアプリケーションを正常に実行できることを確認しました。詳細は付録の通りとなります。

### 5. テスト結果及び考察

COBOL アプリケーションが COBOL データファイルからデータ充填するデータテーブルを使って Crystal Reports 出力を動的に生成できることを確認しました。

以上

## 付録. サンプルアプリケーションの作成・実行

- 1) Start menu > Micro Focus Enterprise Developer > データツール > データ接続 > ADO.NET Connection Editor  
を起動し、SQL Server への ADO データソースを作成します。

※ COBOL クラスより SQL Server にアクセスする際に使用します。

- 2) CsharpWeb.sln という Visual Studio ソリューションを作成します。

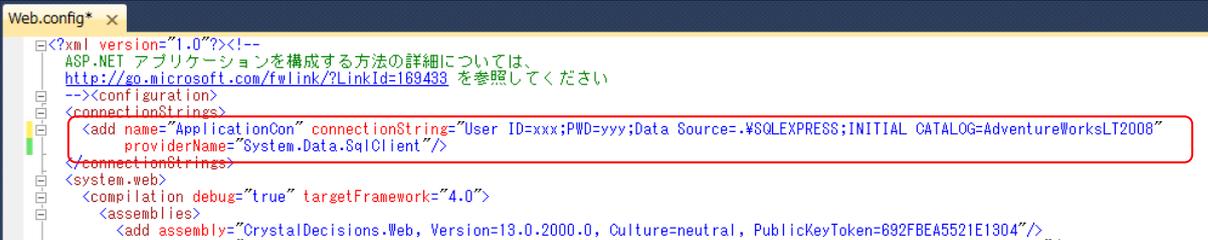
- 3) CsharpWeb.sln 配下に以下の2つのプロジェクトを作成します。

COBOL Class	COBOL プロジェクト > マネージド > クラスライブラリ
CsharpWeb	Visual C# > Web > ASP.NET Web アプリケーション

- 4) CsharpWeb プロジェクト配下の Web.config を開き、ConnectionStrings 要素に「ApplicationCon」という名前の SQL Server への接続文字列を定義します。

※ Default.aspx に定義したデータソースが SQL Server にアクセスする際に使用します。

図 付-1 ConnectionStrings 要素への追加イメージ



```
Web.config* x
<?xml version="1.0"?><!--
ASP.NET アプリケーションを構成する方法の詳細については、
http://go.microsoft.com/fwlink/?LinkId=169438 を参照してください
--><configuration>
  <connectionStrings>
    <add name="ApplicationCon" connectionString="User ID=xxx;PWD=yyy;Data Source=.\SQLEXPRESS;INITIAL CATALOG=AdventureWorksLT2008"
        providerName="System.Data.SqlClient"/>
  </connectionStrings>
  <system.web>
    <compilation debug="true" targetFramework="4.0">
      <assemblies>
        <add assembly="CrystalDecisions.Web, Version=13.0.2000.0, Culture=neutral, PublicKeyToken=692FBEA5521E1304"/>...
```

- 5) CsharpWeb プロジェクトにデータセット DS\_By\_Order\_ID.xsd を新しい項目として追加します。

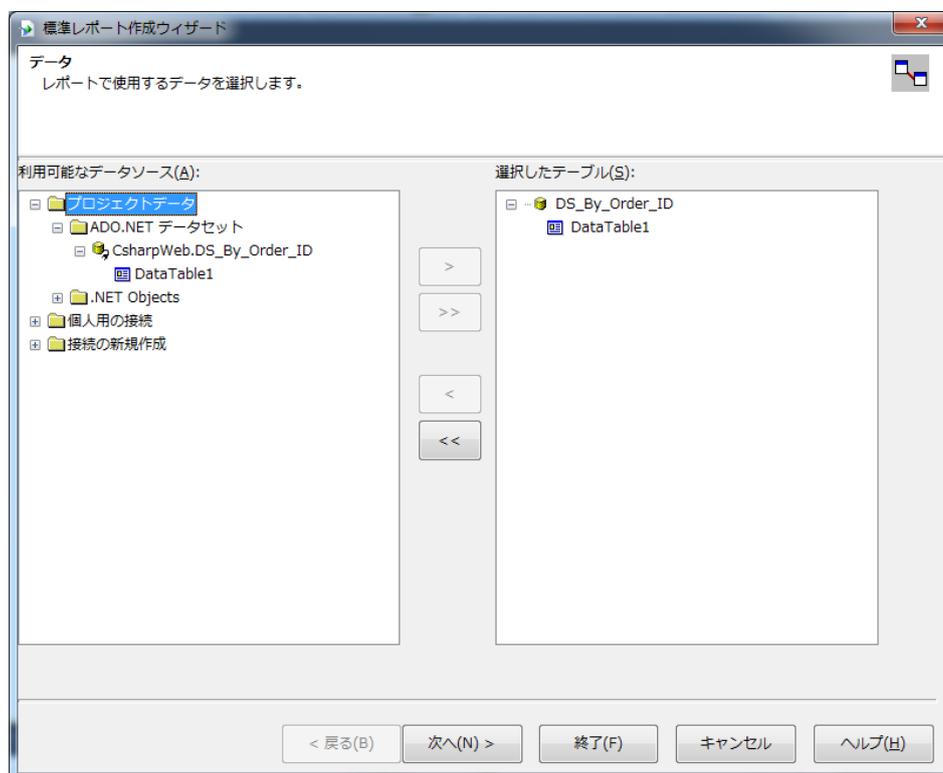
- 6) DS\_By\_Order\_ID.xsd に下表のような列を持つ DataTable を作成します。

表 付-1 DataTable 定義

列名	列定義
Product_Name	System.String
P_Cat_Name	System.String
Company_Name	System.String
Order_Qty	System.Int16
Unit_Price	System.Decimal
Line_Total	System.Decimal

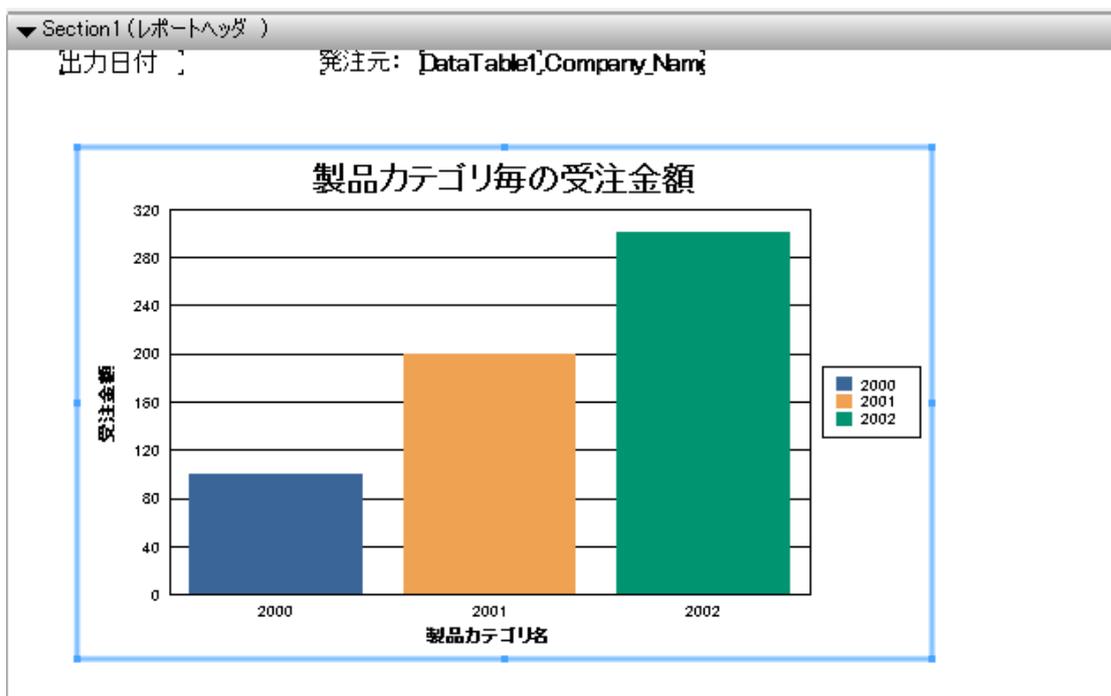
- 7) CsharpWeb プロジェクトに Crystal Reports CR\_By\_Order\_ID.rpt を新しい項目として追加します。
- 8) 標準レポート作成ウィザードにて、上で作成したデータセットに設定した DataTable を選択します。

図 付-2 標準レポート作成ウィザード



- 9) CR\_By\_Order\_ID.rpt のレポートヘッダに、P\_Cat\_Name を横軸に、Line\_Total の合計額を縦軸に持つ並列棒チャートを作成します。

図 付-3 並列棒チャートの設定後のイメージ



- 10) CR\_By\_Order\_ID.rpt に下図のような表を追加します。

図 付-4 表の設定後のイメージ

Section2 (ページヘッダ)				
出力日付 :                      発注元: [DataTable1.Company_Name]				
<b>製品カテゴリ毎の受注金額</b>				
320				
280				
240				
200				
160				
120				
80				
40				
0				
	2000	2001	2002	
	<b>製品カテゴリ名</b>			
	<div style="border: 1px solid black; padding: 2px; display: inline-block;"> <span style="color: blue;">■</span> 2000  <span style="color: orange;">■</span> 2001  <span style="color: green;">■</span> 2002 </div>			
Section3 (詳細)				
製品名	製品カテゴリ	[オーダー数]	[製品価格]	[小計]
Product_Name	P_Cat_Name	[Order_Qty]	[Unit_Price]	[Line_Total]
Section4 (レポートフッタ)				
合計受注金額:				[total]の合計
Section5 (ページフッタ)				
				ページ番号

- 11) 以下のファイルを追加し、アプリケーションロジックを構築。実際のプログラムソースは本報告と共に公開済のものを参照してください。CobFileHandle.cbl の 68 行目の接続先は 1) にて作成したものを指定します。

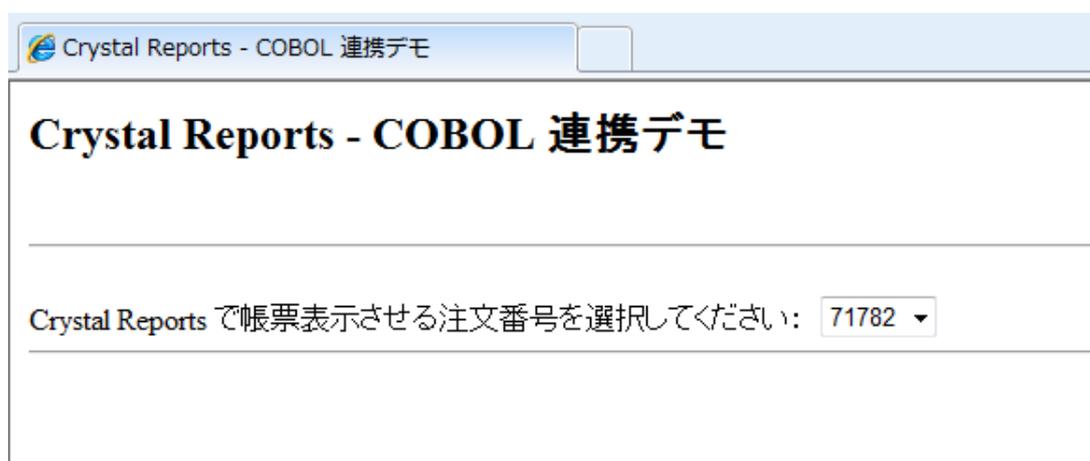
表 付-2 追加したプログラム

プログラムファイル名	言語	プロジェクト
CobFileHandle.cbl	COBOL	COBOL Class
Default.aspx.cs	C#	CsharpWeb
※自動生成		

12) [デバッグ]メニュー > [デバッグ開始] を選択し、デバッグを開始します。

13) 任意の注文番号を選択します。

図 付-5 サンプルアプリケーション起動画面

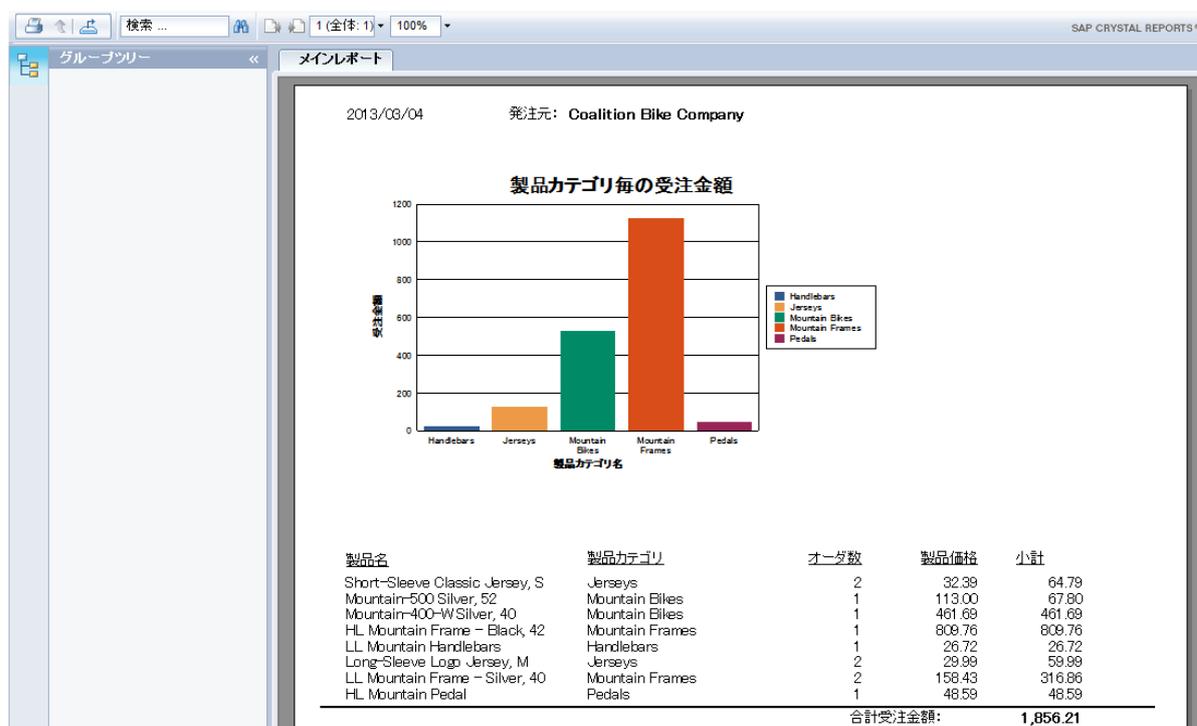


14) 選択した注文番号に応じたデータを基とするレポートが出力されることを確認します。

図 付-6 サンプルアプリケーション実行画面①

Crystal Reports - COBOL 連携デモ

Crystal Reports で帳票表示させる注文番号を選択してください: 71899

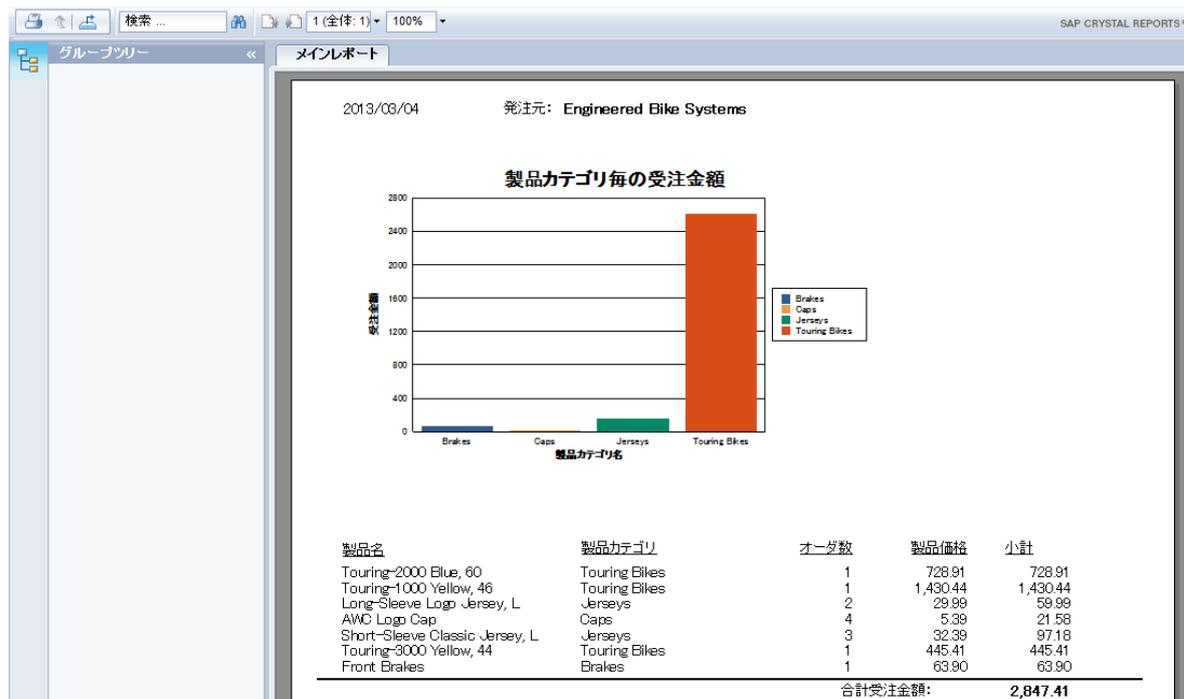


15) 別の注文番号を選択し、その注文番号に応じたレポートが生成されることを確認します。

図 付-7 サンプルアプリケーション実行画面②

Crystal Reports - COBOL 連携デモ

Crystal Reports で帳票表示させる注文番号を選択してください: 71816



以上