



# IBM ILOG Gantt for .NET V4.0

チュートリアル

2009年6月

© Copyright International Business Machines Corporation 1987, 2009.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.



序文	チュートリアル.....	3
<b>Windows Forms</b>	<b>ガント・アプリケーションの作成 .....</b>	<b>5</b>
	ガントチャートの作成 .....	5
	ガント・データ・モデルとガントチャートの関連付け .....	13
	ガントチャートへの XML データ入力.....	14
<b>Web Forms</b>	<b>ガント・アプリケーションの作成.....</b>	<b>19</b>
	ガントチャートの作成 .....	19
	ガント・データ・モデルとガントチャートの関連付け .....	23
	ガントチャートへの XML データ入力.....	23
	ガントチャートとの対話に使用するコントロールの追加 .....	25
	ガント・データ・モデルを変更するコントロールの追加 .....	26
	要求全体を通じた変更の保存 .....	27
スケジュール・チャートと負荷チャートの同期化 ( <b>Windows Forms</b> チュートリアル) .....		<b>29</b>
	スケジュール・チャートの作成 .....	29
	ガント・データ・モデルとスケジュール・チャートの関連付け .....	38
	スケジュール・チャートへの XML データ入力.....	40
	負荷チャートの追加 .....	44

チャートの同期化 .....	48
表示時間範囲の同期化 .....	48
グリッドの同期化 .....	49
チャート区切り線の同期化 .....	50
リソース選択の同期化 .....	51
アプリケーション開始時に予約が表示されるようにする .....	53
<b>カスタム・ガント表示の作成 (Windows Forms チュートリアル) .....</b>	<b>57</b>
アクティビティ・テーブルの作成 .....	57
ガント・モデルとアクティビティ・テーブルの関連付け .....	65
アクティビティ・テーブルへの XML データ入力.....	67
スクロール式アクティビティ・シートの追加 .....	71
アプリケーション開始時に予約が表示されるようにする .....	78
<b>索引 .....</b>	<b>81</b>

# チュートリアル

IBM® ILOG® Gantt for .NET のチュートリアルでは、IBM ILOG Gantt for .NET フレームワークの基本プログラミング領域について、順を追って説明します。各チュートリアルでは、小規模なサンプル・アプリケーションの記述、構築、実行に必要なステップを順番に詳しく説明します。

このセクションでは、以下について取り上げます。

## *Windows Forms* ガント・アプリケーションの作成

Windows® Forms を使用した、簡単な Gantt アプリケーションの作成方法について、順を追って説明します。

## *Web Forms* ガント・アプリケーションの作成

Web Forms を使用した、簡単なガント・アプリケーションの作成方法について、順を追って説明します。

## スケジュール・チャートと負荷チャートの同期化 (*Windows Forms* チュートリアル)

スケジュール・チャートと負荷チャートを同期化する方法について、順を追って説明します。

## カスタム・ガント表示の作成 (*Windows Forms* チュートリアル)

アクティビティ・テーブルと 2 つのアクティビティ・シートから成るカスタム・ガントチャート表示の作成方法について、順を追って説明します。

## 関連セクション

### スタートアップ

IBM ILOG Gantt for .NET フレームワークの実用的な基本情報と概要を示します。

# Windows Forms ガント・アプリケーションの作成

このチュートリアルでは、簡単な Windows® Forms ガント・アプリケーションの作成方法について説明します。

## 関連項目

[ガントチャートの作成 | ガント・データ・モデルとガントチャートの関連付け | ガントチャートへの XML データ入力](#)

---

## ガントチャートの作成

IBM® ILOG® Gantt for .NET では、ガントチャートを用いて、スケジューリング情報の表示を制御できます。ガントチャートは、次の 3 つの部分に分かれています。

- ◆ テーブルビュー (GanttTable のインスタンス) には、データ・モデルからのアクティビティ情報が表示されます。
- ◆ ガント・シート (クラス GanttSheet のインスタンス) には、時間尺に沿ったアクティビティの配置が表示されます。
- ◆ 時間尺 (クラス TimeScale のインスタンス) はズーム可能です。

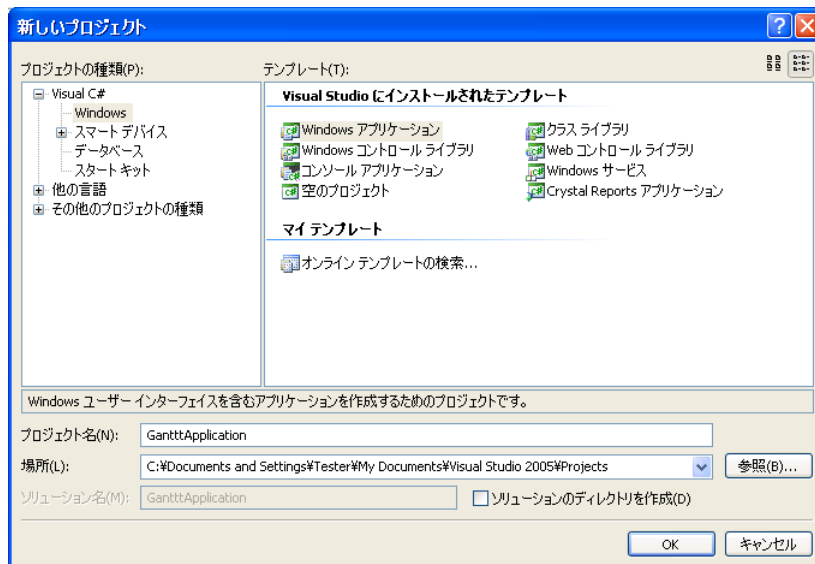
標準の調整可能な区切り線によって、左側と右側が区切られます。

Gantt アプリケーションを作成するには、以下の手順を実行します。

### 1. Windows® Forms タイプのプロジェクトを作成します。

[ファイル] メニューの [新規作成] をクリックし、次に [プロジェクト] をクリックします。

[新しいプロジェクト] ウィンドウが表示されます。



2. 左側のウィンドウで、[Visual C#] > [Windows] をクリックします。

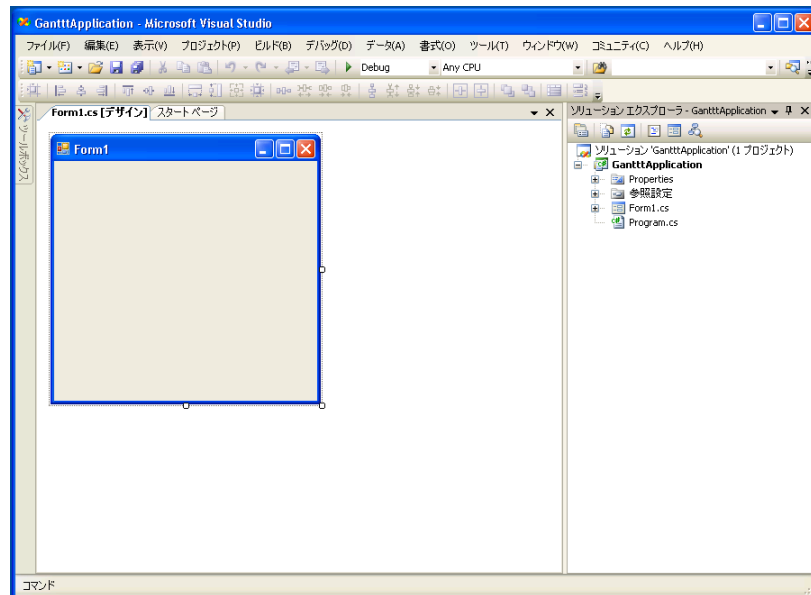
3. 右側のウィンドウで [Windows アプリケーション] をクリックします。

4. プロジェクトに GanttApplication という名前を付けます。

5. プロジェクトを C:\Gantt に配置します。

Visual Studio®.NET によって、Form1 という名前のフォームが作成され、デフォルトで、ファイル **Form1.cs** に保存されます。





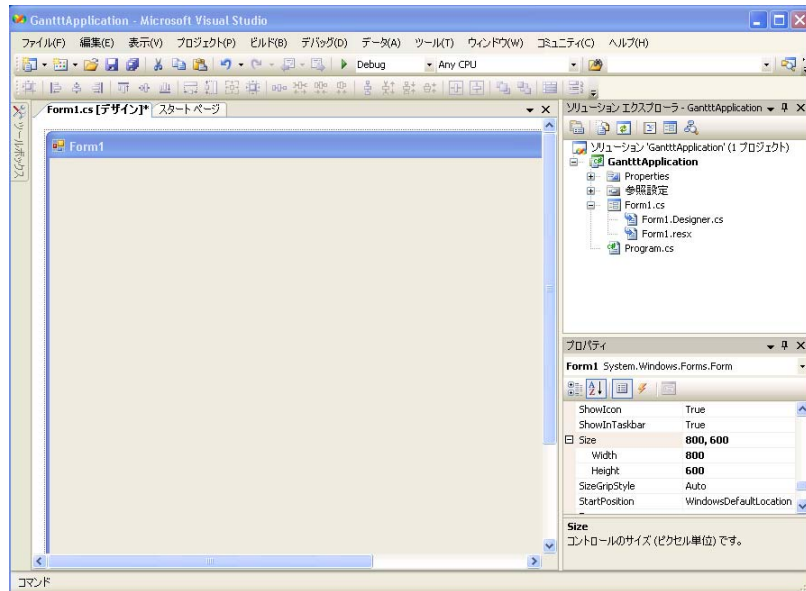
6. ウィンドウのサイズを変更します。

メニューバーで[表示]>[プロパティ・ウィンドウ]とクリックし、プロパティ・エクスプローラで[配置]>[Size]とクリックします。プロパティを展開し、次の値を入力します。

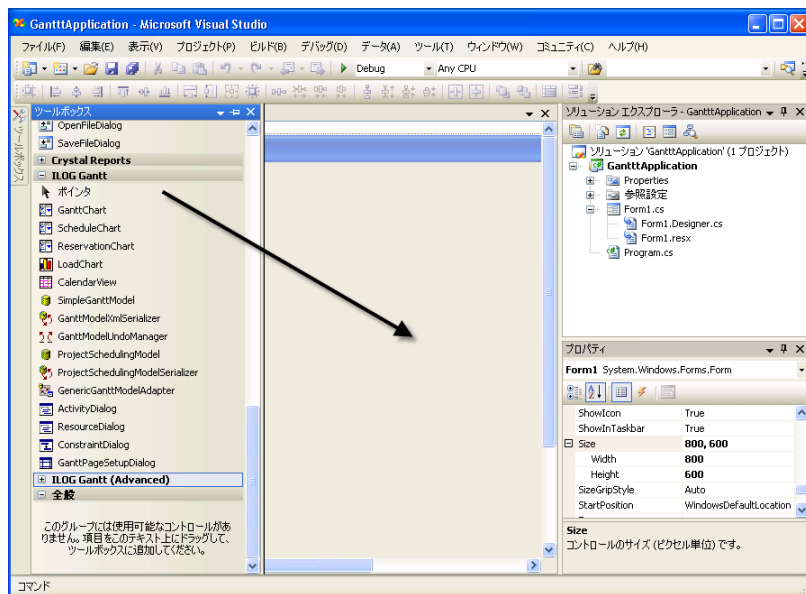
幅 : 800

高さ : 600

## Windows Forms ガント・アプリケーションの作成

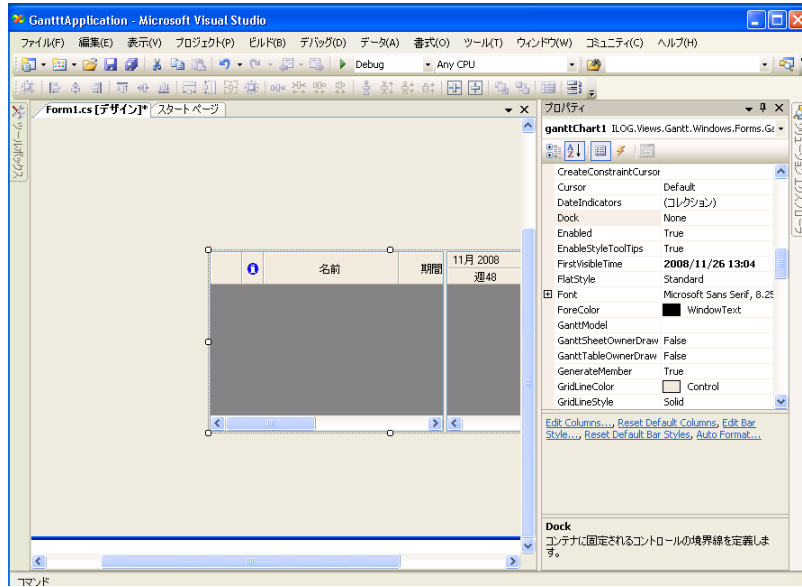


7. IBM ILOG Gantt for .NET ツールボックスで、**GanttChart** オブジェクトを選択して Form1 までドラッグします。

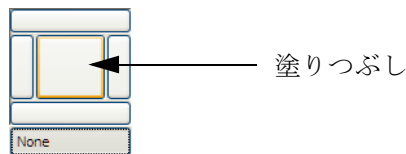


8. ガントチャートがフォーム全体に表示されるよう、Dock プロパティを変更します。

Form1 のガントチャートをクリックし、プロパティ・エクスプローラで [ 配置 ] > [ Dock ] とクリックします。

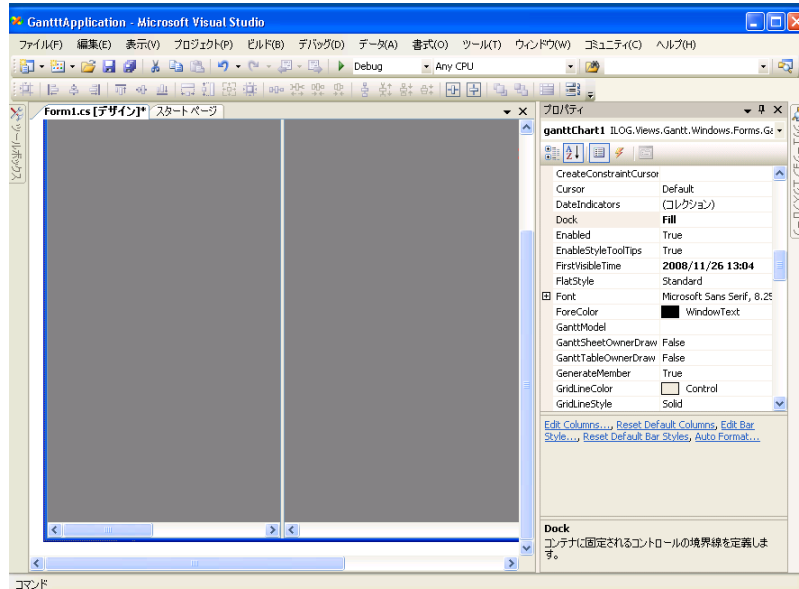


Dock プロパティの隣の矢印をクリックすると、次のコントロールが表示されます。



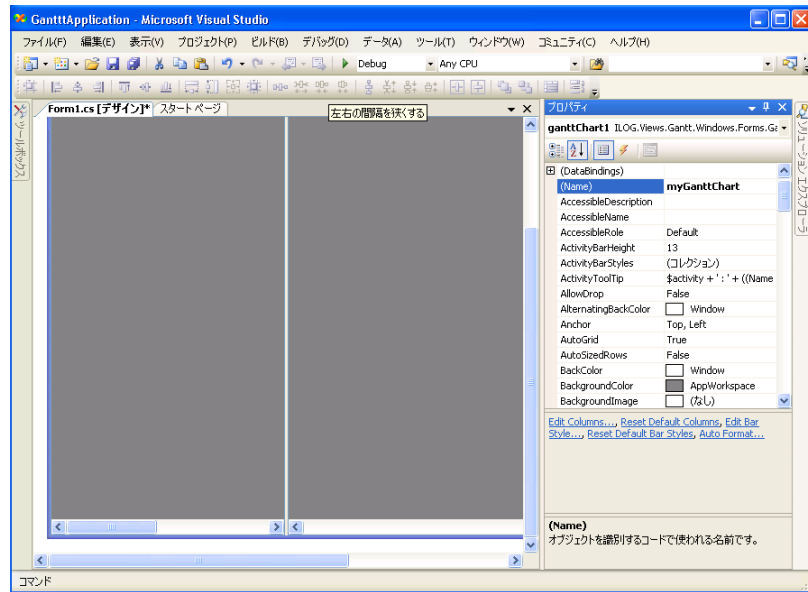
9. 塗りつぶしオプションを適用するエリアを選択します。

Form1 は次のように表示されます。



10. ガントチャートの名前を変更します。

Form1 のガントチャートをクリックします。プロパティ・エクスプローラで[デザイン] > [Name] とクリックし、**myGanttChart** という新しい名前を入力します。

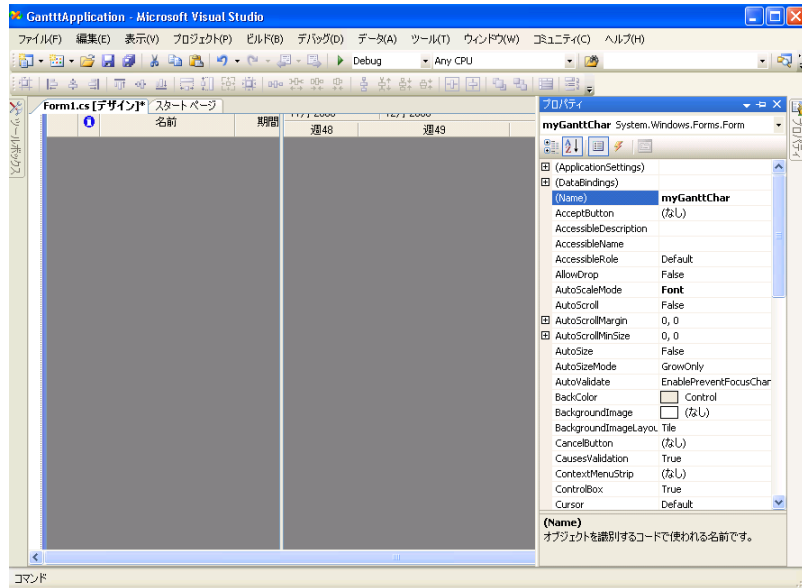


ガントチャートの外観は、ユーザーのニーズに合わせて、完全にカスタマイズできます。

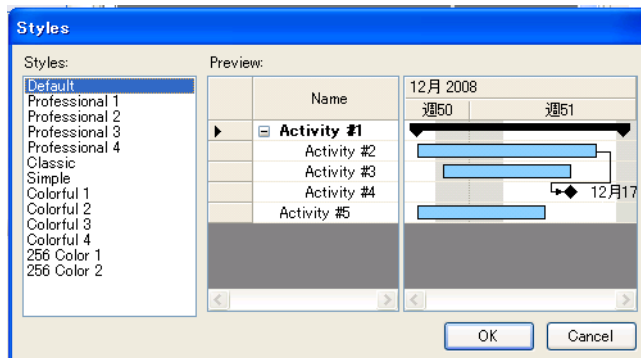
**11. 作成したガントチャートのスタイルを定義します。**

Form1 のガントチャートを右クリックし、ショートカット・メニューの **[AutoFormat]** をクリックするか、プロパティ・エクスプローラの下部に表示される **[AutoFormat]** をクリックします。

## Windows Forms ガント・アプリケーションの作成



[スタイル] ウィンドウが表示されます。



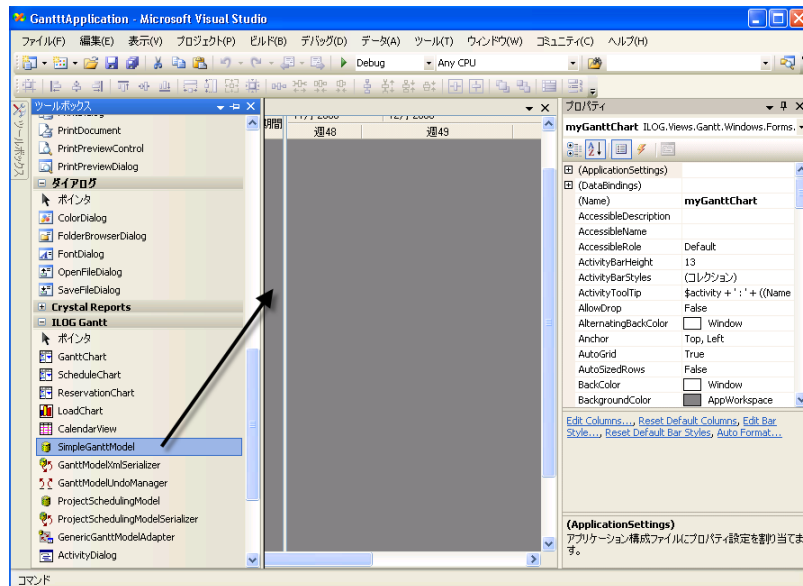
スタイルを選択し、[OK] をクリックします。選択したスタイルが、ガントチャートに自動的に反映されます。

12. ファイル **Form1.cs** を保存します。

## ガント・データ・モデルとガントチャートの関連付け

ガント・データ・モデルには、チャートに表示されるスケジューリング情報が含まれます。以下では、簡単なガント・データ・モデルをガントチャートに結合する方法について説明します。

1. IBM® ILOG® Gantt for .NET ツールボックスで、**SimpleGanttModel** オブジェクトを選択して Form1 までドラッグします。



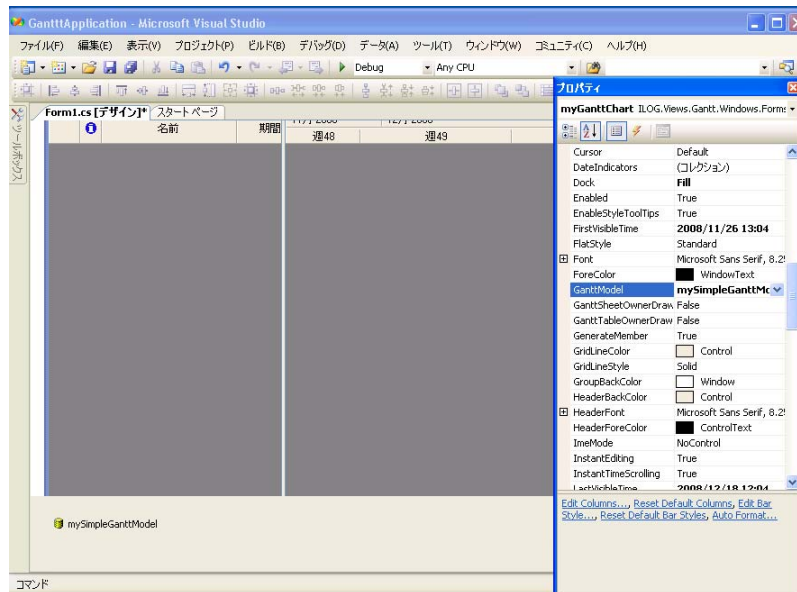
2. この簡単なガント・モデルの名前を変更します。

Form1 の **[SimpleGanttModel1]** オブジェクトをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**mySimpleGanttModel** という新しい名前を入力します。

3. **mySimpleGanttModel** と **myGanttChart** を関連付けます。

Form1 のガントチャートをクリックし、プロパティ・エクスプローラで [データ] > **[GanttModel]** とクリックします。

GanttModel プロパティの横に表示されるドロップダウン・リストの矢印をクリックし、**[mySimpleGanttModel]** をクリックします。



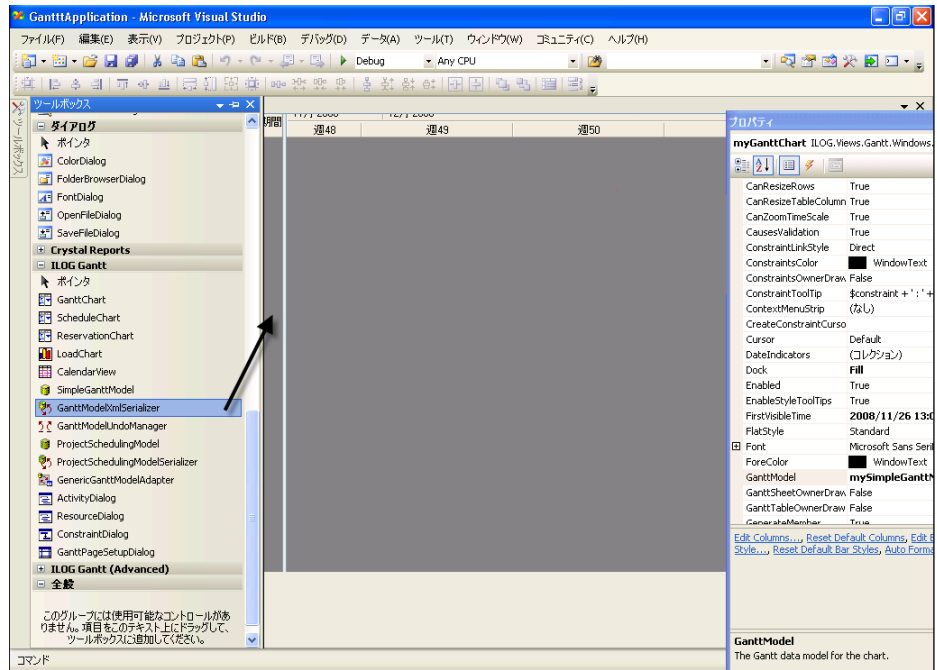
### ガントチャートへの XML データ入力

IBM® ILOG® Gantt for .NET では、Scheduling Data Exchange Language (SDXL) ファイルのスケジューリング・データをシリアル化・非シリアル化できます。これには、**GanttModelXmlSerializer** クラスを使用します。

SDXL ファイルからのスケジューリング・データを非シリアル化した後、この情報をガントチャートへの入力に使用するには、以下の手順を実行します。

1. IBM ILOG Gantt for .NET ツールボックスで、**GanttModelXmlSerializer** オブジェクトを選択して Form1 までドラッグします。





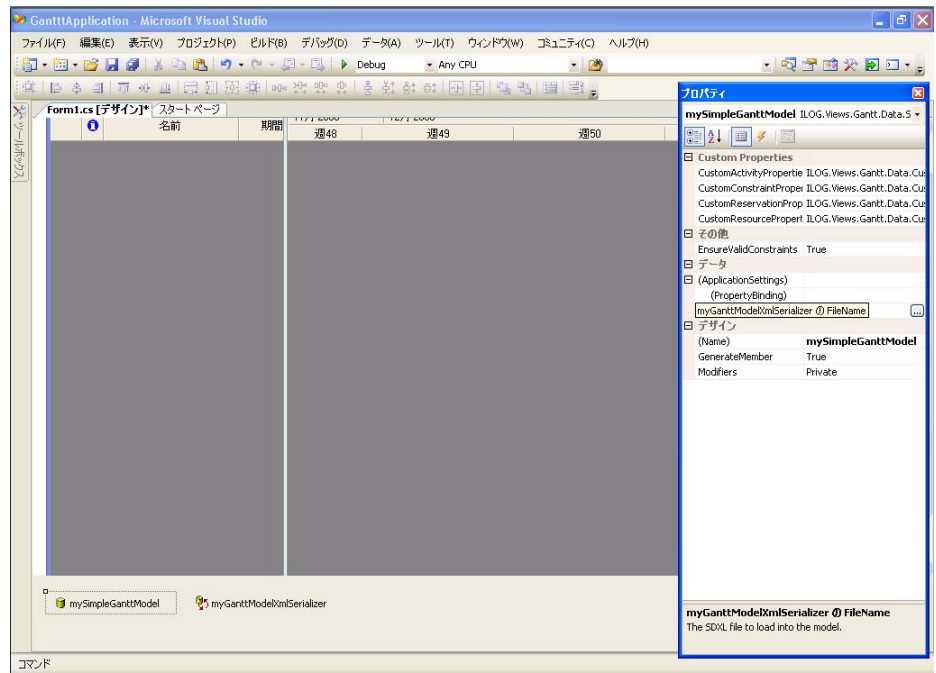
2. このガント・モデル・シリアライザの名前を変更します。

Form1 の [ganttModelXmlSerializer1] オブジェクトをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**myGanttModelXmlSerializer** という新しい名前を入力します。

3. [mySimpleGanttModel] をクリックします。

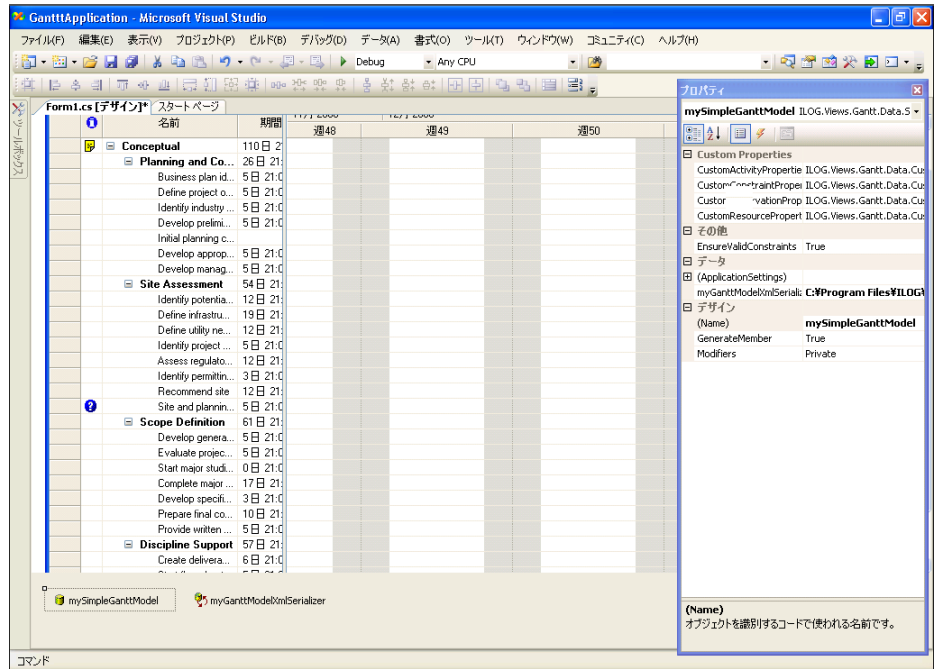
[myGanttModelXmlSerializer の FileName] という名前の新規プロパティが作成されています。

## Windows Forms ガント・アプリケーションの作成



4. プロパティ [**myGanttModelXmlSerializer**] の [**FileName**] をクリックし、隣の3つの点が表示されているボタンをクリックします。ガントチャートに入力するSDXL ファイルをクリックします。

**注:** サンプル SDXL ファイルは <install-dir>\Data にあります。



SDXL ファイルに含まれるデータがガントチャートに表示されます。

5. ファイル **Form1.cs** を保存します。
6. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

Form1 という名前の Gantt アプリケーションが、次のように表示されます。





# Web Forms ガント・アプリケーションの作成

このチュートリアルでは、簡単な Web Forms ガント・アプリケーションの作成方法について説明します。

## 関連項目

[ガントチャートの作成](#) | [ガント・データ・モデルとガントチャートの関連付け](#) | [ガントチャートへのXML データ入力](#) | [ガントチャートとの対話に使用するコントロールの追加](#) | [ガント・データ・モデルを変更するコントロールの追加](#) | [要求全体を通じた変更の保存](#)

---

## ガントチャートの作成

IBM® ILOG® Gantt for .NET では、ガントチャートを用いて、スケジューリング・データの表示を制御できます。ガントチャートは、次の 3 つの部分に分かれています。

- ◆ テーブル・ビュー (GanttTable のインスタンス) には、データ・モデルからのアクティビティ情報が表示されます。
- ◆ ガント・シート (クラス GanttSheet のインスタンス) には、時間尺に沿ったアクティビティの配置が表示されます。
- ◆ 時間尺 (クラス TimeScale のインスタンス) はズーム可能です。

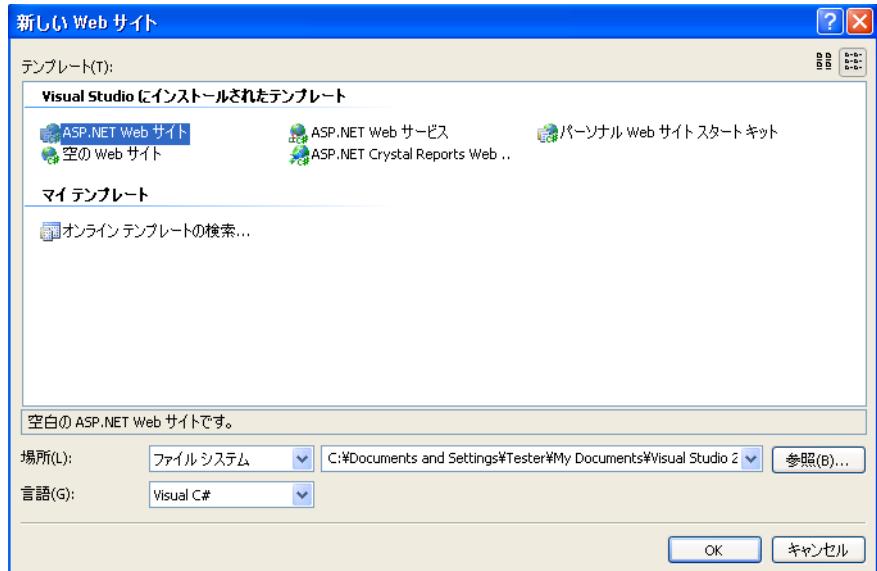
標準の調整可能な区切り線によって、左側と右側が区切られます。

Gantt アプリケーションを作成するには、以下の手順を実行します。

1. ASP.NET Web アプリケーション・タイプのプロジェクトを作成します。

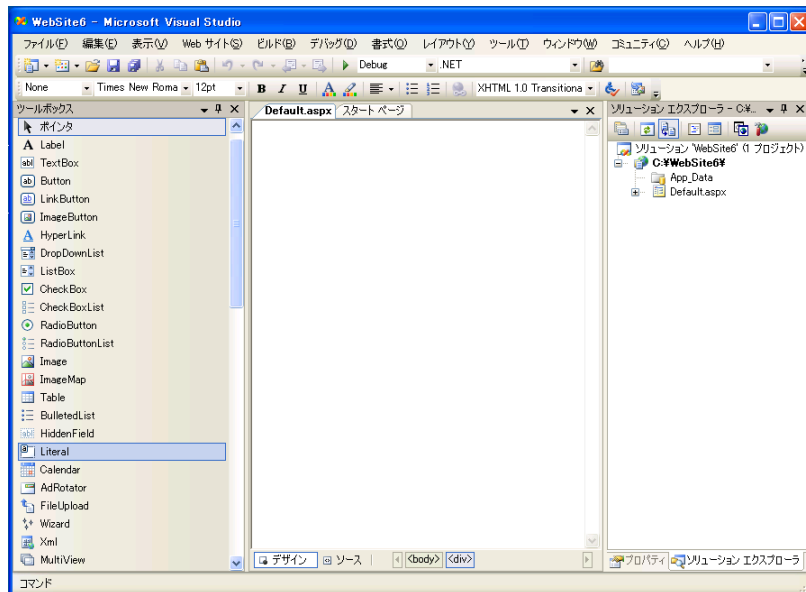
[ファイル] メニューの [新規作成] をクリックし、次に [Web サイト] をクリックします。

[新しいプロジェクト] ウィンドウが表示されます。

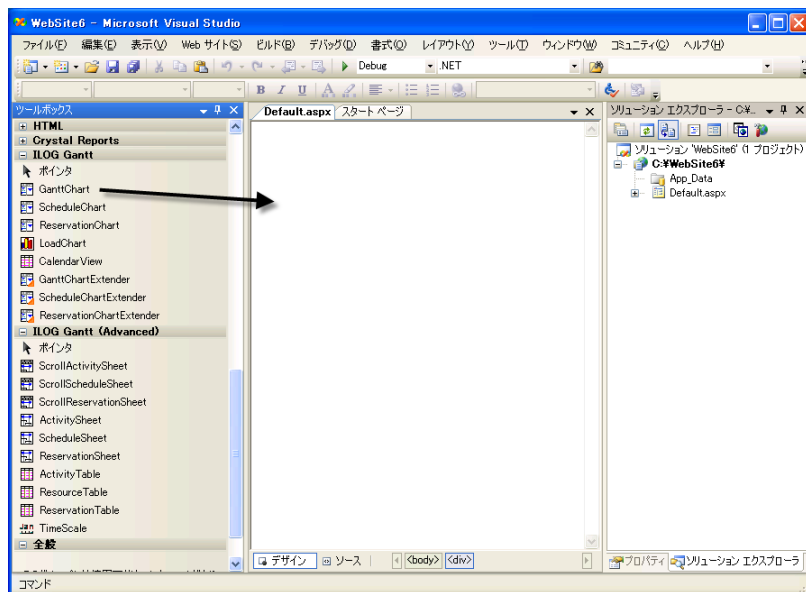


2. [ASP.NET Web サイト] をクリックします。
3. [格納場所] テキスト・ボックス内をクリックします。
4. プロジェクトに **C:\WebGanttApplication** という名前を付けます。
5. [OK] ボタンをクリックします。

Visual Studio®.NET によって、空のフォームが作成され、デフォルトで、ファイル **Default.aspx** に保存されます。

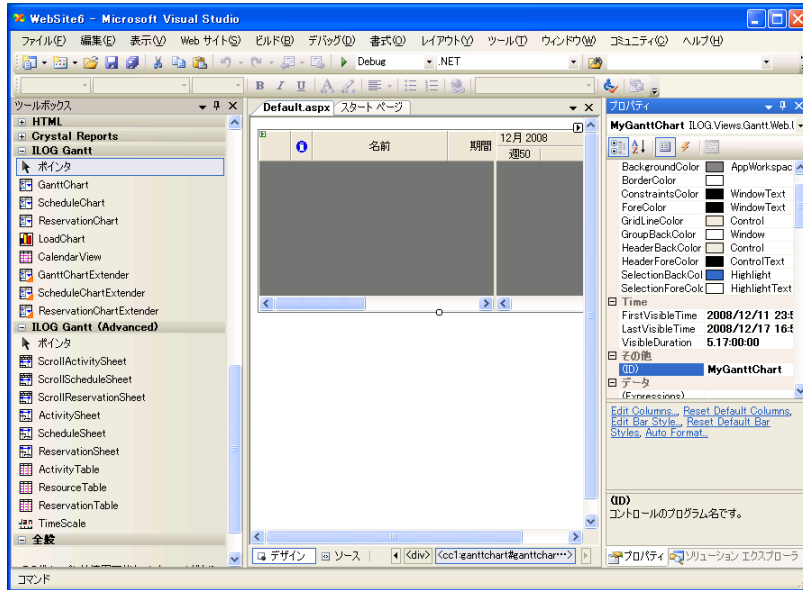


6. IBM ILOG Gantt for .NET ツールボックス・タブで、**GanttChart** オブジェクトを選択して Web Form までドラッグします。



7. ガントチャートの名前を変更します。

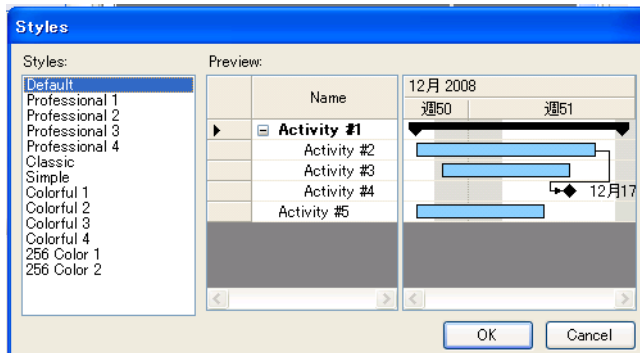
Web Form 内のガントチャートをクリックします。プロパティ・エクスプローラで **[Misc] > [ID]** とクリックし、**MyGanttChart** という新しい名前を入力します。



8. ガントチャートのサイズを拡大します。

9. 作成したガントチャートのスタイルを定義します。

[デフォルト] のガントチャートを右クリックし、ショートカット・メニューの **[AutoFormat]** をクリックするか、プロパティ・エクスプローラの下部に表示される **[AutoFormat]** をクリックします。[スタイル] ウィンドウが表示されます。





スタイルを選択し、**[OK]** をクリックします。選択したスタイルが、ガントチャートに自動的に反映されます。

10. ファイル **Default.aspx** を保存します。

## ガント・データ・モデルとガントチャートの関連付け

ガント・データ・モデルには、チャートに表示されるスケジューリング・データが含まれます。以下では、簡単なガント・データ・モデルをガントチャートに結合する方法について説明します。

1. **Default.aspx.cs** ファイルを開き、次の変数宣言文を追加します。

```
private ILOG.Views.Gantt.Data.SimpleGanttModel ganttModel;
```

2. **OnInit** メソッドを上書きして、ガント・モデルを作成します。

```
protected override void OnInit(EventArgs e)
{
    ganttModel = new ILOG.Views.Gantt.Data.SimpleGanttModel();
    base.OnInit(e);
}
```

3. **OnLoad** メソッドを上書きして、ガントチャートのガント・モデルを設定します。

```
protected override void OnLoad(EventArgs e)
{
    MyGanttChart.GanttModel = ganttModel;
    base.OnLoad(e);
}
```

## ガントチャートへの XML データ入力

IBM® ILOG® Gantt for .NET では、Scheduling Data Exchange Language (SDXL) ファイルのスケジューリング・データをシリアル化・非シリアル化できます。これには、**GanttModelXmlSerializer** クラスを使用します。

SDXL ファイルからのスケジューリング・データを非シリアル化した後、この情報をガントチャートへの入力に使用するには、以下の手順を実行します。

1. **Default.aspx.cs** ファイルを開き、次の変数宣言文を追加します。

```
private ILOG.Views.Gantt.Data.GanttModelXmlSerializer ganttSerializer;
```

2. **OnInit** メソッドを変更して、シリアライザを作成します。

```
protected override void OnInit(EventArgs e)
{
    ganttModel = new ILOG.Views.Gantt.Data.SimpleGanttModel;
    ganttSerializer = new ILOG.Views.Gantt.Data.GanttModelXmlSerializer();
    base.OnInit(e);
}
```

3. **OnLoad** メソッドを変更し、SDXL ファイルをガント・モデルに読み込みます。

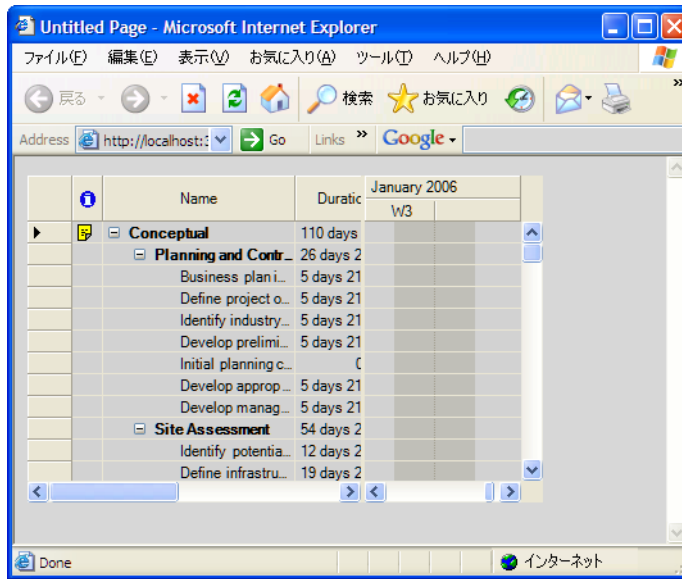
```
protected override void OnLoad(EventArgs e)
{
    MyGanttChart.GanttModel = ganttModel;
    ganttSerializer.SetFileName(ganttModel, filename);
    base.OnLoad(e);
}
```

filename を、SDXL ファイルの保存場所に置き換えます。

**注:** サンプル SDXL ファイルは <install-dir>\Data にあります。

4. ファイルをすべて保存します。
5. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

**WebGanttApplication** という名前の Gantt アプリケーションが、次のように表示されます。



ガントのスクロールバーを使うと、タイム・ラインに沿って移動できます。列を選択したり、親列の [ 展開 / 折りたたみ ] ボタンをクリックして列を展開したり、折りたたむこともできます。

## ガントチャートとの対話に使用するコントロールの追加

このセクションでは、ガントチャートと対話に使用する標準 Web Forms コントロールの追加方法について説明します。

特に、ボタンを使用して、ガントチャートのズーム・レベルを変更する方法を取り上げます。

1. [Web Form ツールボックス] タブのボタンをクリックし、Web Form にドラッグします。
2. プロパティ・エクスプローラで [ 表示 ] > [Text] とクリックし、Zoom In という新規テキストを入力します。
3. ボタンをダブルクリックすると、イベント・ハンドラが追加され、ソース・コード・エディタが表示されます。ボタン・イベント・ハンドラに、GanttChart の ZoomIn メソッドの呼び出しを追加します。

```
private void Button1_Click(object sender, System.EventArgs e)
{
    MyGanttChart.ZoomIn();
}
```

[Default.aspx] タブをクリックして、デザイン・ビューに戻ります。

1. [Web Form ツールボックス] タブのボタンをクリックし、Web Form にドラッグします。
2. プロパティ・エクスプローラで [表示] > [Text] とクリックし、Zoom Out という新規テキストを入力します。
3. ボタンをダブルクリックすると、イベント・ハンドラが追加され、ソース・コード・エディタが表示されます。ボタン・イベント・ハンドラに、GanttChart の ZoomOut メソッドの呼び出しを追加します。

```
private void Button2_Click(object sender, System.EventArgs e)
{
    MyGanttChart.ZoomOut();
}
```

[Default.aspx] タブをクリックして、デザイン・ビューに戻ります。

4. ファイル Default.aspx を保存します。
5. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

これで、ガントチャートのスクロール・バー、[Zoom In (拡大)]、[Zoom Out (縮小)] ボタンを使用して、タイム・ラインに沿って移動できるようになります。

---

## ガント・データ・モデルを変更するコントロールの追加

このセクションでは、ガントチャートで読み込んだガント・データ・モデルを変更する標準 Web Forms コントロールの追加方法について説明します。

特に、ガント・データ・モデルにアクティビティを追加する方法を取り上げます。

1. [Web Form ツールボックス] タブのボタンをクリックし、Web Form にドラッグします。
2. プロパティ・エクスプローラで [表示] > [Text] とクリックし、New Activity という新規テキストを入力します。
3. ボタンをダブルクリックすると、イベント・ハンドラが追加され、ソース・コード・エディタが表示されます。ボタン・イベント・ハンドラに、次のコードを追加します。

```
private void Button3_Click(object sender, System.EventArgs e)
{
    int index = MyGanttChart.NewActivity(
        null,
        "New Activity",
```

```

        MyGanttChart.VisibleTimeInterval);
    if (index != -1)
        MyGanttChart.EnsureRowVisible(index);
}

```

**[Default.aspx]** タブをクリックして、デザイン・ビューに戻ります。

4. コマンド **[デバッグ]** > **[デバッグ開始]** を使用して、アプリケーションを実行します。

**[New Activity (新規アクティビティ)]** ボタンをクリックすると、新しいアクティビティが追加されます。

**[New Activity]** ボタンをもう一度クリックしても、アクティビティが1つしか追加されないことを確認します。これは、ラウンドトリップの度に、SDXL ファイルからモデルが再読み込みされるためです。次のセクションでは、ラウンド・トリップ間でモデルに行った変更を維持する方法について説明します。

## 要求全体を通じた変更の保存

クライアントによってページが要求される度、これらの要求を処理するために、ページの新規インスタンスが作成されます。これは、コントロールやデータを含む、ページ内のあらゆる要素が再び作成されることを意味します。ASP.NET のページは、要求が行われる度、複数のフェーズ間を移行します。次の要求が行われるまで、ページやコントロールが、ある種の状態を保っているように見えるのはこのためです。以下では、**LoadViewState** フェーズと **SaveViewState** フェーズという、2つのページ・ライフ・サイクルについて、詳しく取り上げます。

コントロール・ライフ・サイクルの **LoadViewState** フェーズでは、要求に含まれるデータを用いて、コントロールが初期化されます。このデータは **ViewState** と呼ばれます。

コントロール・ライフ・サイクルの **SaveViewState** フェーズでは、コントロールの状態が **ViewState** に保存されることにより、次の要求で利用可能になります。

このセクションでは、**ViewState** にガント・データ・モデルを保存する方法について説明します。実際アプリケーションでは、**ViewState** にモデルを保存するのではなく、データベースを使用しますが、これについては、本チュートリアルでは取り上げません。

1. ロード・イベント・ハンドラのコードを変更し、ページの初回要求時にのみ SDXL ファイルが読み込まれるように設定します。

これを行うには、**Web Form** の空白部分をダブルクリックします。ソース・コード・エディタが表示されます。既存のコードを次のコードに置き換えます。

```

protected override void OnLoad(EventArgs e)
{
    MyGanttChart.GanttModel = ganttModel;
}

```

```
        if (!IsPostBack)
            ganttSerializer.SetFileName(ganttModel, filename);
        base.OnLoad(e);
    }
```

filename を、SDXL ファイルの保存場所に置き換えます。

**注:** サンプル SDXL ファイルは <install-dir>\Data にあります。

2. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

**[New Activity]** ボタンをクリックすると、モデルには追加されたアクティビティのみが含まれています。これは、SDXL ファイルが最初のページの要求時のみ読み込まれ、ポストバック要求時には読み込まれないためです。

3. Web Form を右クリックしてコード・ビューに切り換え、[コードの表示] をクリックします。
4. 次のコードを追加して、**SaveViewState** メソッドを上書きします。

```
protected override object SaveViewState() {
    System.IO.StringWriter writer = new System.IO.StringWriter();
    ganttSerializer.Serialize(writer, ganttModel);
    ViewState["GanttModelDescription"] = writer.ToString();
    writer.Close();
    return base.SaveViewState();
}
```

上記のコードにより、ガント・データ・モデルが XML 文字列にシリアル化され、ページ **ViewState** に保存されます。

5. 次のコードを追加して、**LoadViewState** メソッドを上書きします。

```
protected override void LoadViewState(object savedState) {
    base.LoadViewState(savedState);
    string xml = ViewState["GanttModelDescription"] as string;
    if (xml != null) {
        System.IO.StringReader reader = new System.IO.StringReader(xml);
        ganttSerializer.Deserialize(reader, ganttModel);
        reader.Close();
    }
}
```

上記のコードにより、**ViewState** からスケジューリング・データの XML 記述が取得され、ガント・データ・モデルにデータが入力されます。

6. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

**[New Activity]** ボタンを複数クリックし、要求と要求の間に行われた変更が保存されていることを確認します。

# スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)

このチュートリアルでは、スケジュール・チャートと負荷チャートを同期化する方法を順番に説明します。最初にチャートを2つ作成し、次に、両チャートのコンポーネントを同期化します。

## 関連項目

[スケジュール・チャートの作成 | ガント・データ・モデルとスケジュール・チャートの関連付け | スケジュール・チャートへのXMLデータ入力 | 負荷チャートの追加 | チャートの同期化 | アプリケーション開始時に予約が表示されるようにする](#)

---

## スケジュール・チャートの作成

IBM® ILOG® Gantt for .NET では、スケジュール・チャートを用いて、スケジュールリング情報の表示を制御できます。スケジュール・チャートは、次の3つの部分に分かれています。

- ◆ 左側のリソース・テーブル (クラス `GanttTable` のインスタンス) には、データ・モデルのリソース情報が表示されます。
- ◆ 右側のガント・シート (クラス `GanttSheet` のインスタンス) には、リソース予約が表示されます。

- ◆ ガント・シートの上部に、ズーム可能な時間尺 (クラス TimeScale のインスタンス) が表示されます。

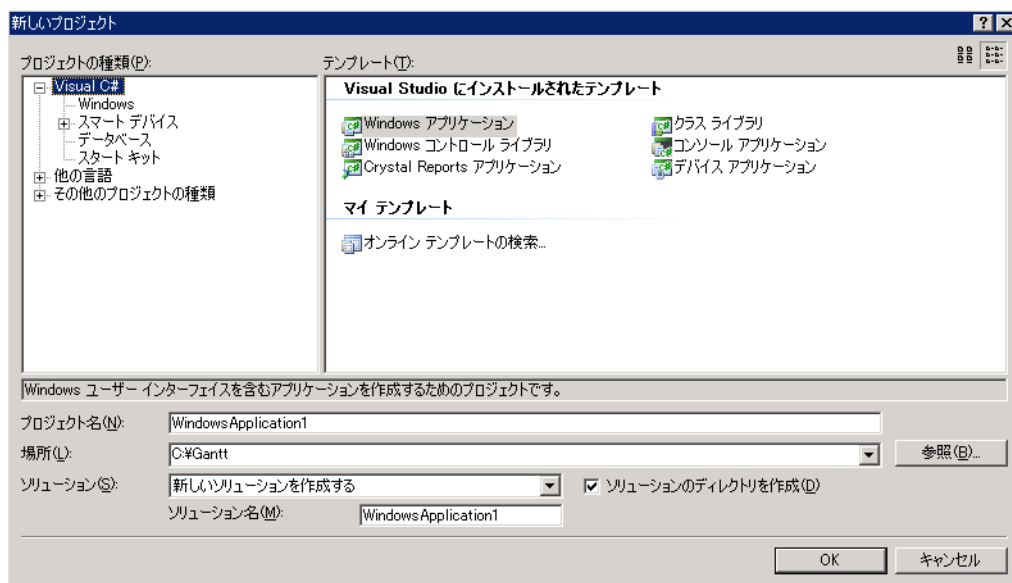
標準の調整可能な区切り線によって、左側と右側が区切られます。

スケジュール・チャートを作成するには、以下の手順を実行します。

1. WindowsR Forms タイプのプロジェクトを作成します。

[ファイル] メニューの [新規作成] をクリックし、次に [プロジェクト] をクリックします。

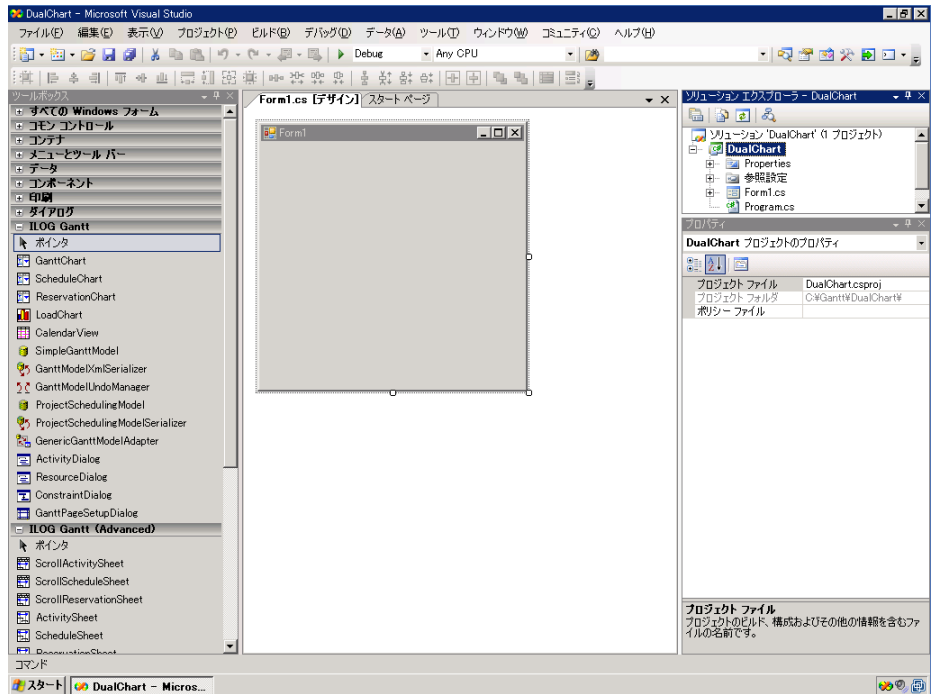
[新しいプロジェクト] ウィンドウが表示されます。



2. 左側のウィンドウで、[Visual C#] > [Windows] をクリックします。
3. 右側のウィンドウで [Windows アプリケーション] をクリックします。
4. プロジェクトに DualChart という名前を付けます。
5. プロジェクトを C:\Gantt に配置します。

Visual Studio®.NET によって Form1 が作成され、デフォルトで **Form1.cs** という名前のファイルに保存されます。





6. ウィンドウのサイズを変更します。

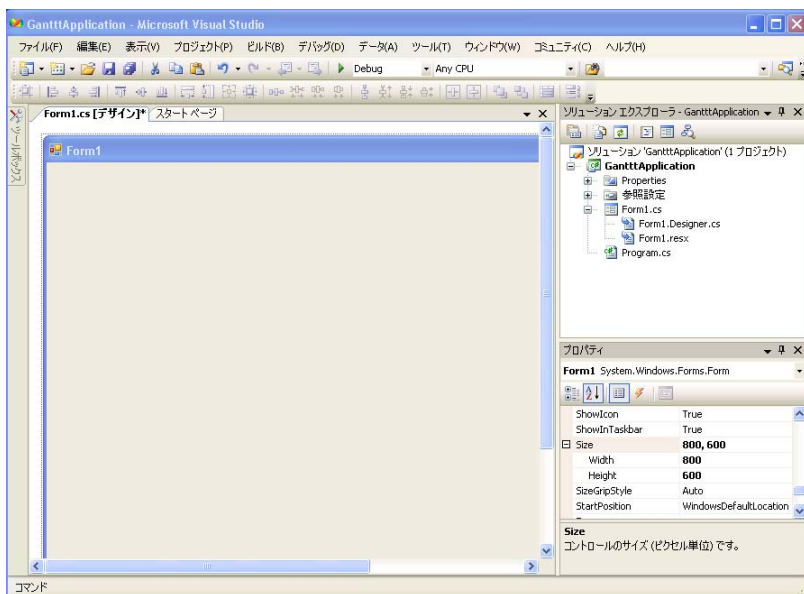
Form1 をクリックし、ウィンドウの右下にある [プロパティ] タブをクリックします。

プロパティ・エクスプローラで [配置] > [Size] とクリックします。プロパティを展開し、次の値を入力します。

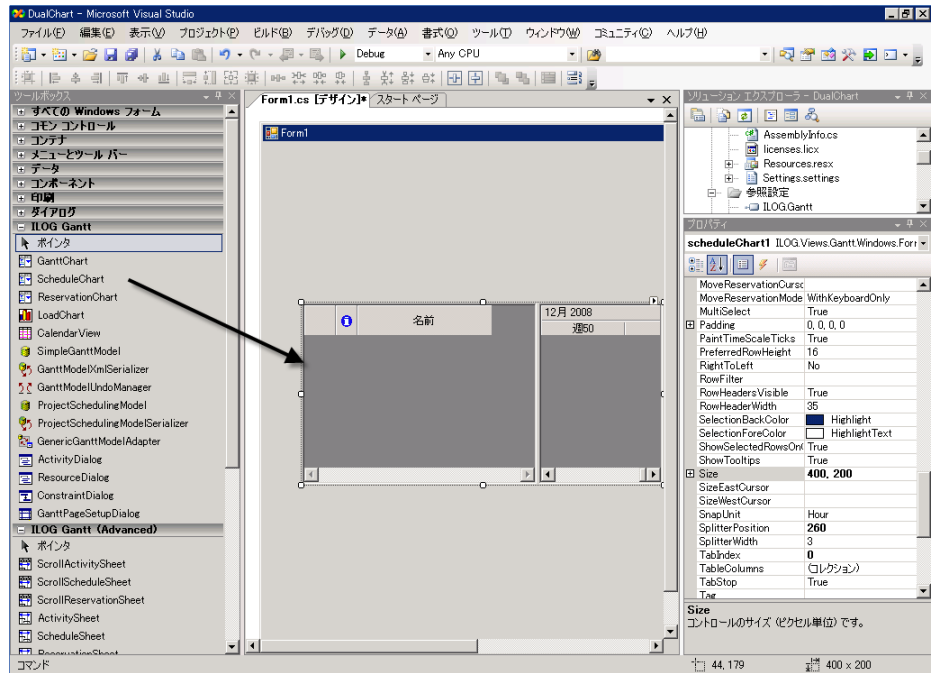
幅 : 800

高さ : 600

## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)



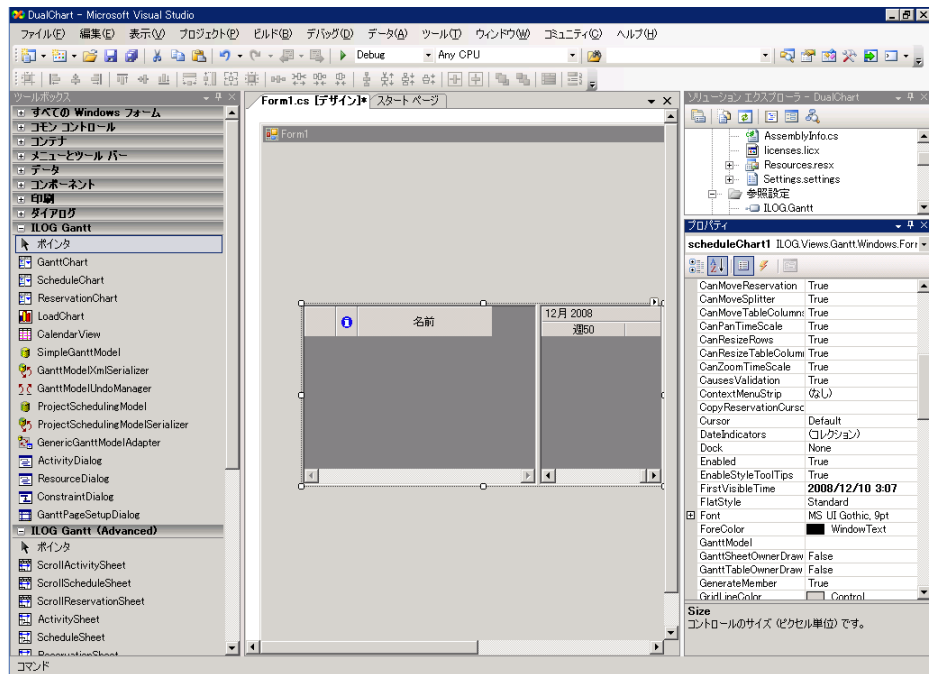
7. IBM ILOG Gantt for .NET ツールボックスで、**ScheduleChart** オブジェクトを選択して Form1 までドラッグします。



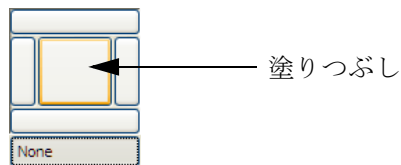
8. スケジュール・チャートがフォーム全体に表示されるよう、Dock プロパティを変更します。

Form1 のスケジュール・チャートをクリックし、プロパティ・エクスプローラで [ 配置 ] > [Dock] とクリックします。

## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)

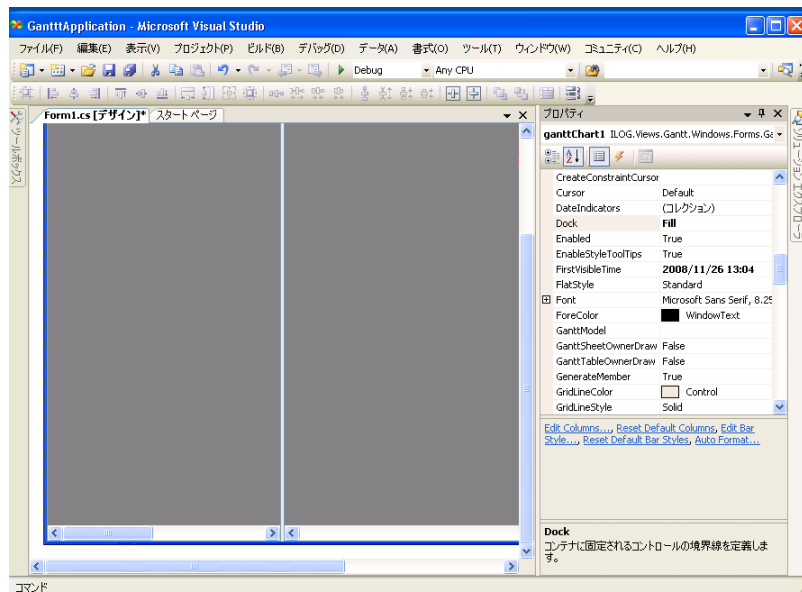


9. [Dock] プロパティの隣の矢印をクリックすると、次のボックスが表示されます。



10. 塗りつぶしオプションを適用するエリアを選択します。

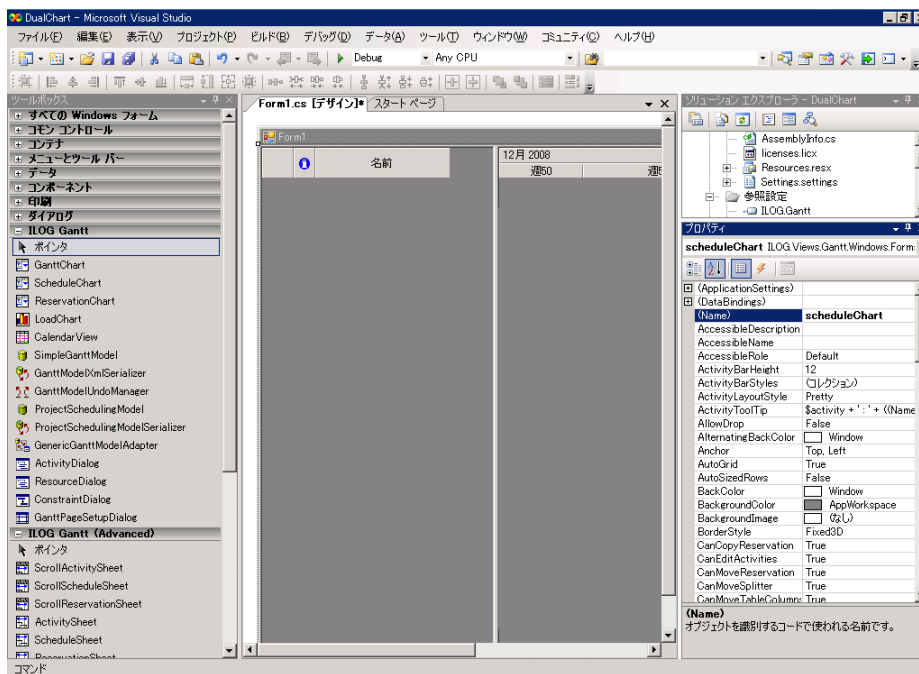
これで、次の図のように、フォーム全体にスケジュール・チャートが表示されました。



#### 11. スケジュール・チャートの名前を変更します。

Form1 のスケジュール・チャートをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**scheduleChart** という新しい名前を入力します。

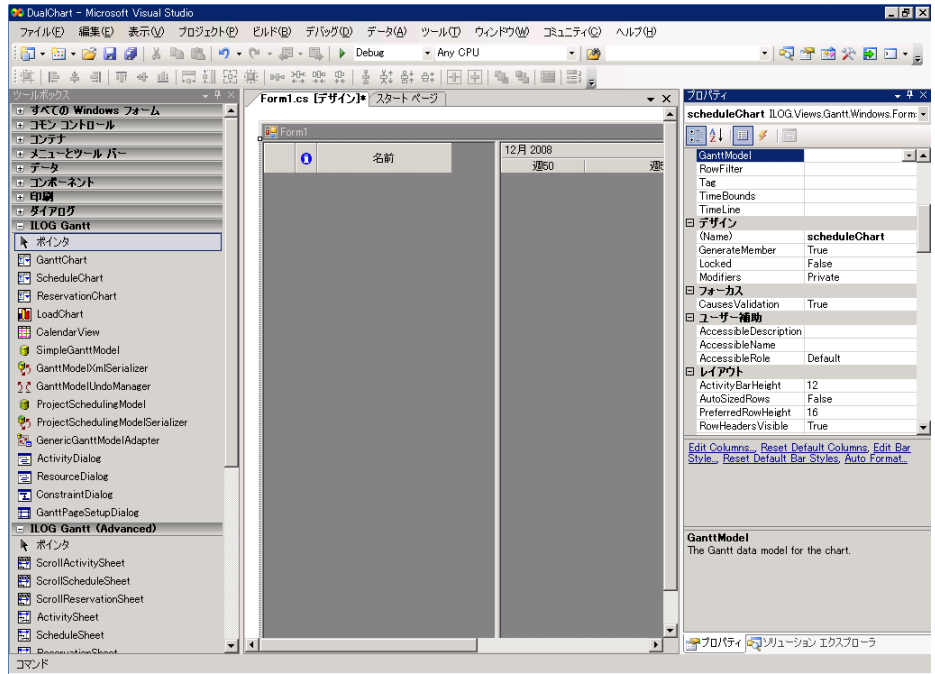
## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)



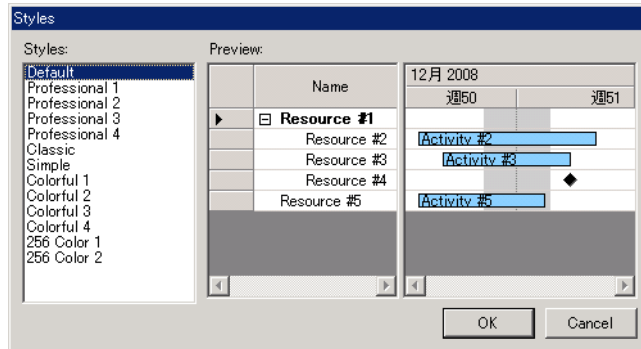
スケジュール・チャートの外観は、ユーザーのニーズに合わせて、完全にカスタマイズできます。

### 12. 作成したスケジューリング・チャートのスタイルを定義します。

スケジュール・チャートを右クリックし、コンテキスト・メニューの [AutoFormat] をクリックするか、プロパティ・エクスプローラの下部に表示される [AutoFormat] をクリックします。



[スタイル] ウィンドウが表示されます。



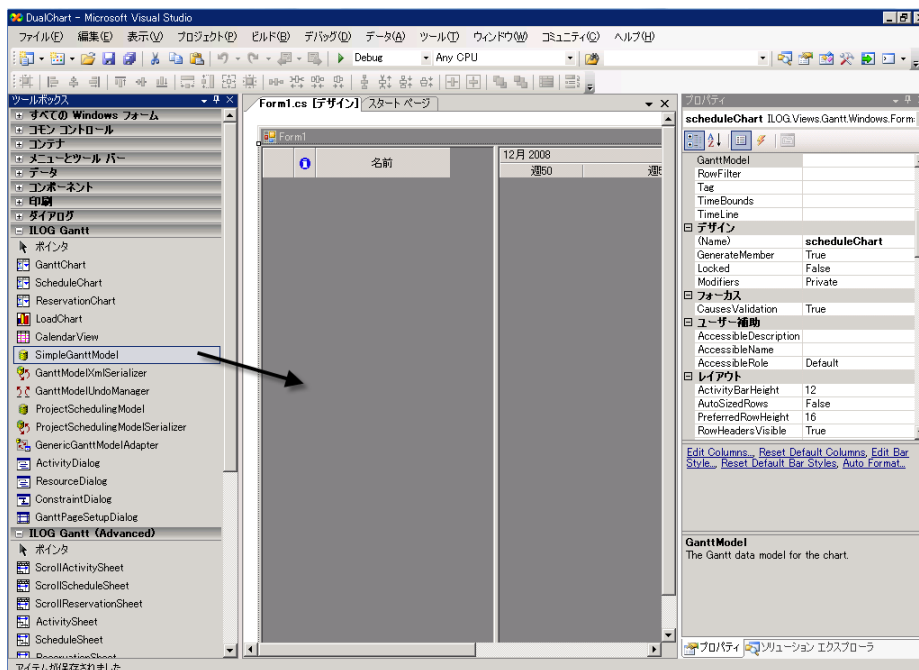
スタイルを選択し、[OK] をクリックします。選択したスタイルが、スケジュール・チャートに自動的に反映されます。

13. ファイル **Form1.cs** を保存します。

## ガント・データ・モデルとスケジュール・チャートの関連付け

データ・モデルには、スケジュール・チャートに表示されるスケジュール情報が含まれます。以下では、簡単なガント・データ・モデルをスケジュール・チャートに結合する方法について説明します。

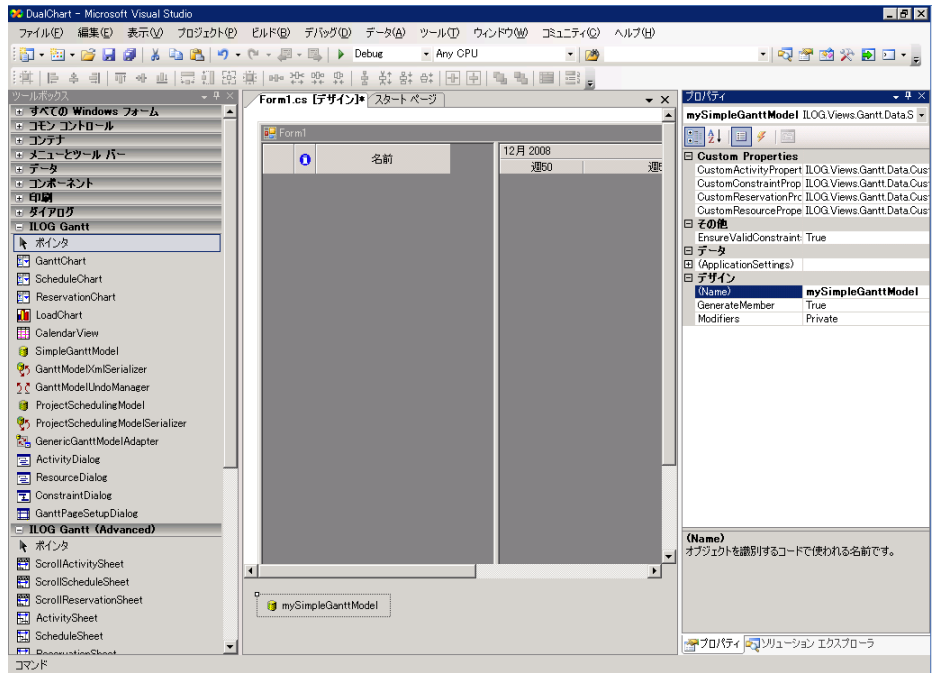
1. IBM® ILOG® Gantt for .NET ツールボックスで、**SimpleGanttModel** オブジェクトを選択して Form1 までドラッグします。



2. **simpleGanttModel** オブジェクトの名前を変更します。

Form1 の [SimpleGanttModel1] オブジェクトをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**mySimpleGanttModel** という名前を入力します。

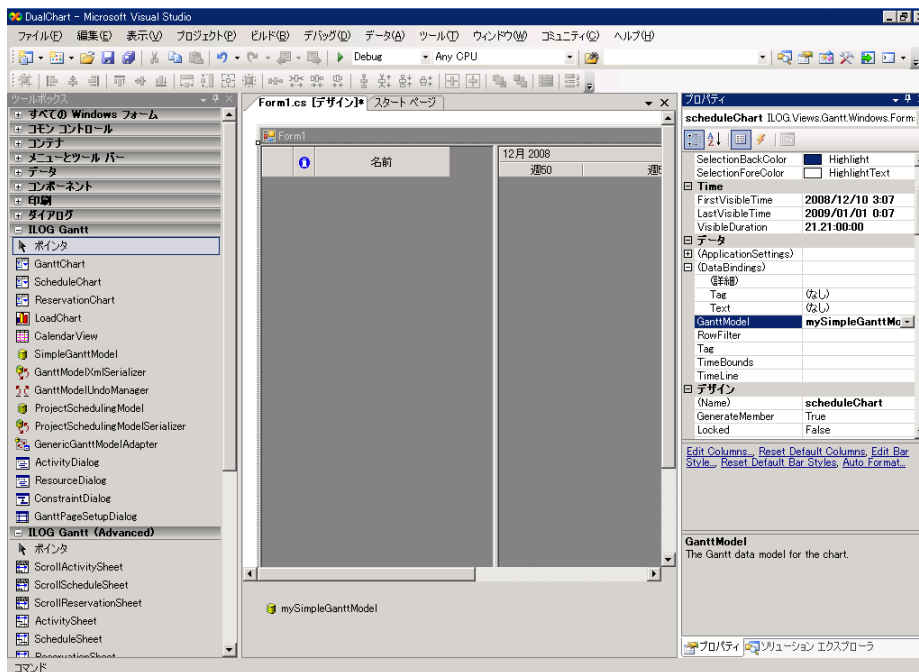




### 3. mySimpleGanttModel と scheduleChart を関連付けます。

Form1 のスケジュール・チャートをクリックし、プロパティ・エクスプローラで [データ] > [GanttModel] とクリックします。

GanttModel プロパティの横に表示されるドロップダウン・リストの矢印をクリックし、[mySimpleGanttModel] をクリックします。

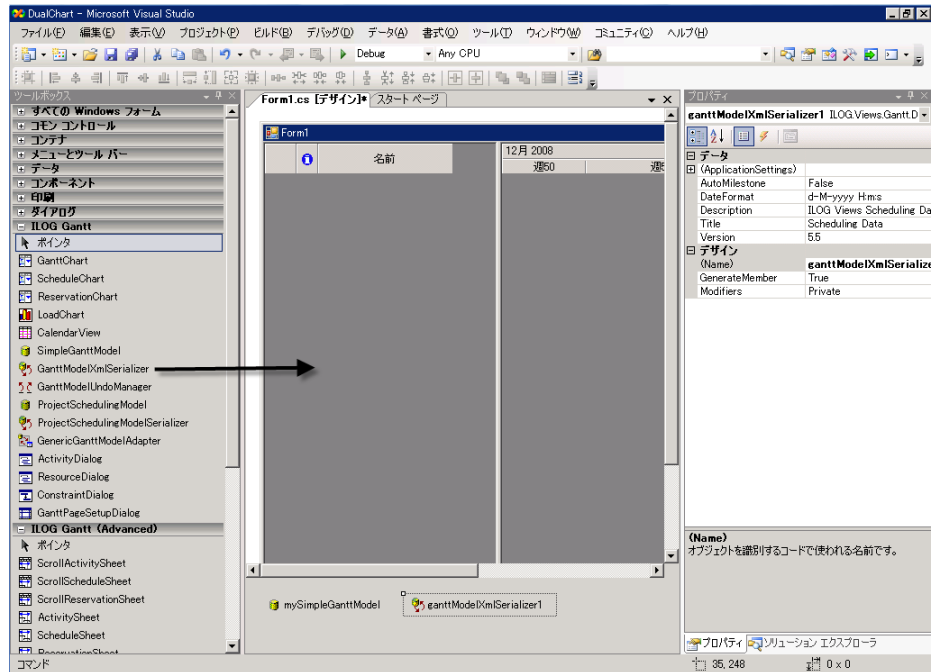


## スケジュール・チャートへの XML データ入力

IBM® ILOG® Gantt for .NET では、Scheduling Data Exchange Language (SDXL) ファイルのスケジューリング・データをシリアル化・非シリアル化できます。これには、**GanttModelXmlSerializer** クラスを使用します。

SDXL ファイルからのスケジューリング・データを非シリアル化した後、この情報をスケジュール・チャートへの入力に使用するには、以下の手順を実行します。

1. IBM ILOG Gantt for .NET ツールボックスで、**GanttModelXmlSerializer** オブジェクトを選択して Form1 までドラッグします。

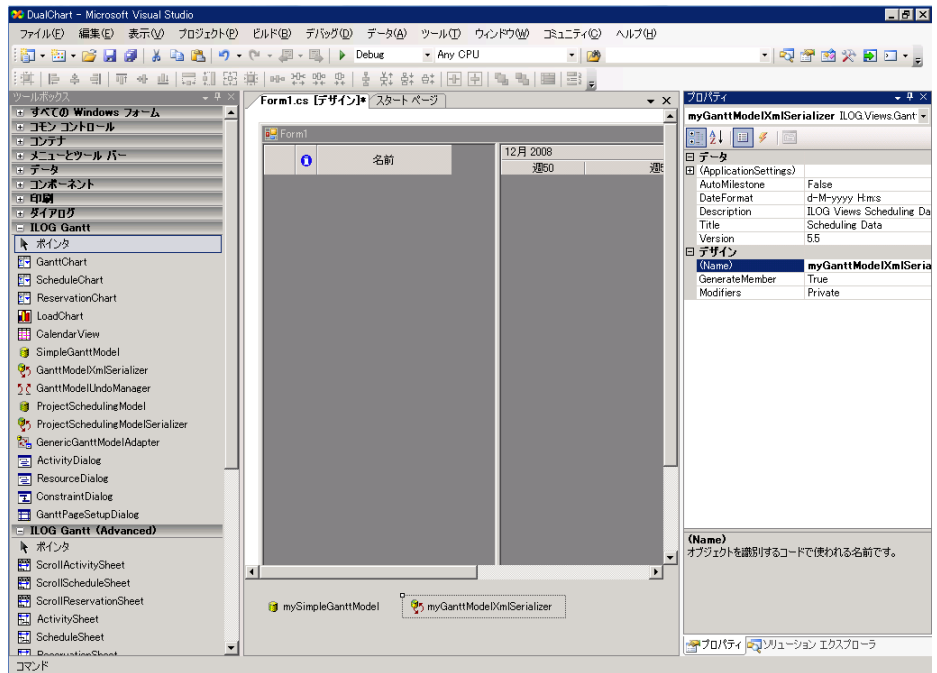


2. このガント・モデル・シリアライザの名前を変更します。

Form1 の **ganttModelXmlSerializer1** オブジェクトをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**myGanttModelXmlSerializer** という新しい名前を入力します。

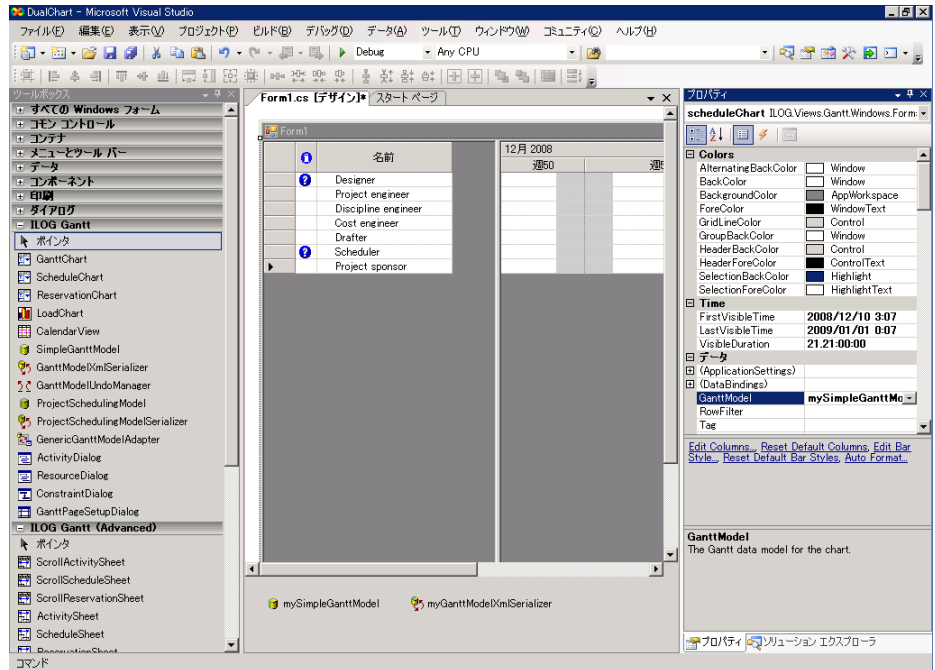
3. [**mySimpleGanttModel**] をクリックします。  
[**myGanttModelXmlSerializer** の **FileName**] という名前の新規プロパティが作成されています。

## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)



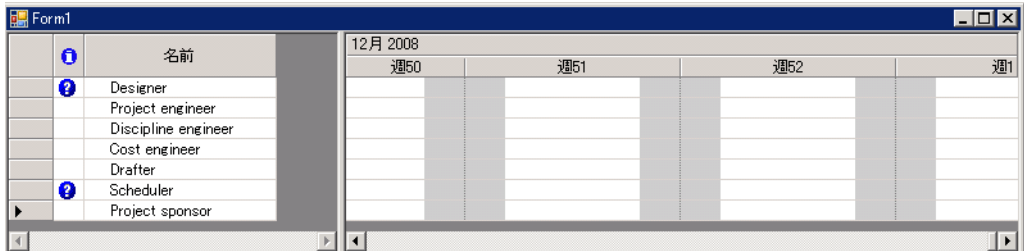
4. プロパティ [ganttModelXmlSerializer] の [FileName] をクリックし、隣の3つの点が表示されているボタンをクリックします。スケジュール・チャートに入力する SDXL ファイルをクリックします。

**注:** サンプル SDXL ファイルは <install-dir>\Data にあります。



**Engineering.sdxl** ファイルに含まれるデータがスケジュール・チャートに表示されます。

5. ファイル **Form1.cs** を保存します。
6. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。  
アプリケーションが次のように表示されます。



7. 実行したアプリケーションを終了し、Form1.cs に戻ります。

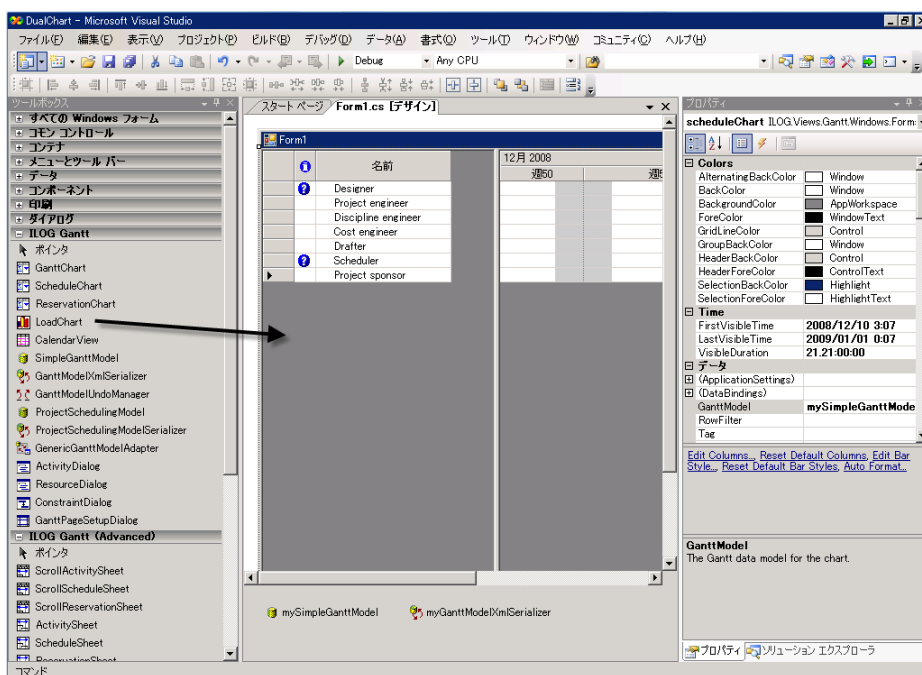
## 負荷チャートの追加

ここでは、負荷チャートを作成し、スケジュール・チャートに関連付ける方法について説明します。

負荷チャートには、時系列に沿って、リソースの負荷がグラフ表示されます。グラフの x 軸が時間、y 軸がリソース負荷を示します。

負荷チャートを作成するには、以下の手順を実行します。

1. IBM® ILOG® Gantt for .NET ツールボックスで、**LoadChart** オブジェクトを選択して Form1 までドラッグします。

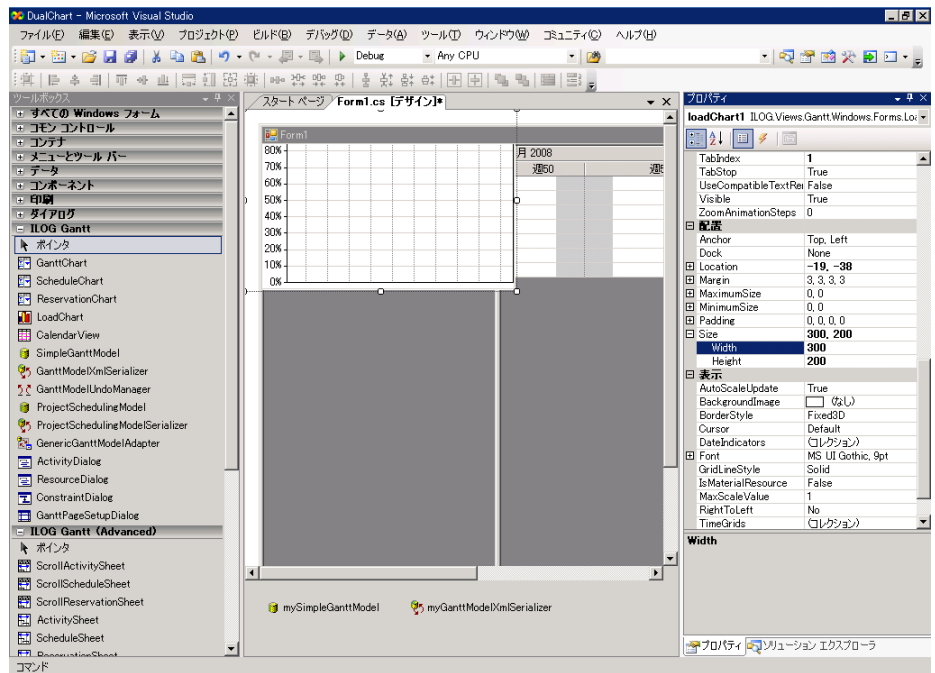


2. 負荷チャートの名前を変更します。

Form1 の負荷チャートをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**loadChart** という新しい名前を入力します。

3. 負荷チャートの高さを変更します。

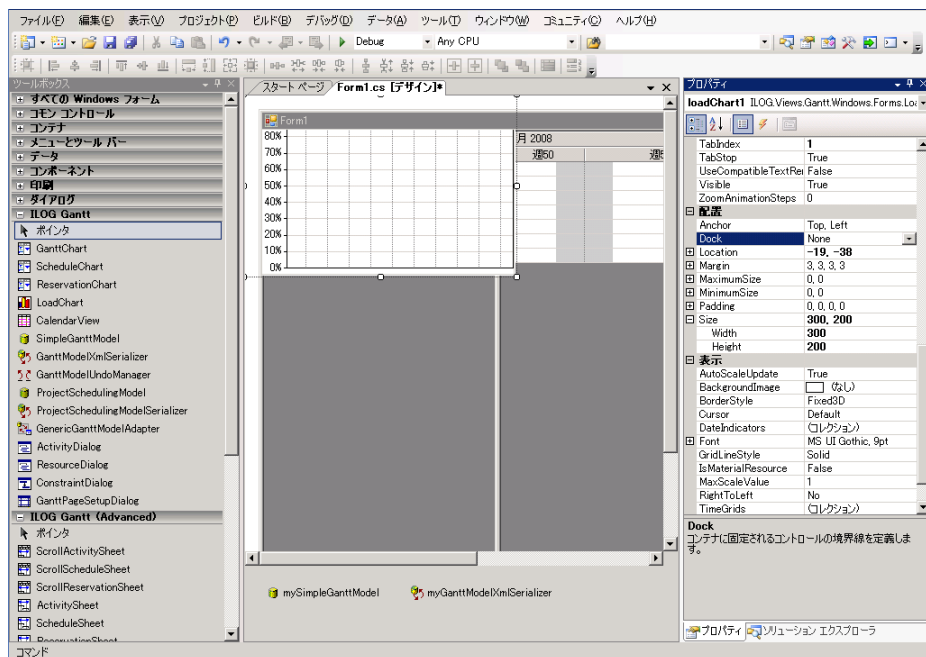
プロパティ・エクスプローラで [配置] > [Size] > [Height] とクリックし、**200** と入力します。



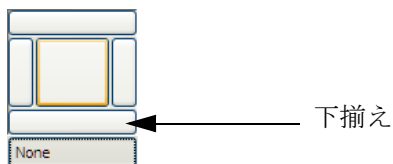
4. 負荷チャートがフォーム全体に表示されるよう、Dock プロパティを変更します。

負荷チャートをクリックし、プロパティ・エクスプローラで [ 配置 ] > [ Dock ] とクリックします。

## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)

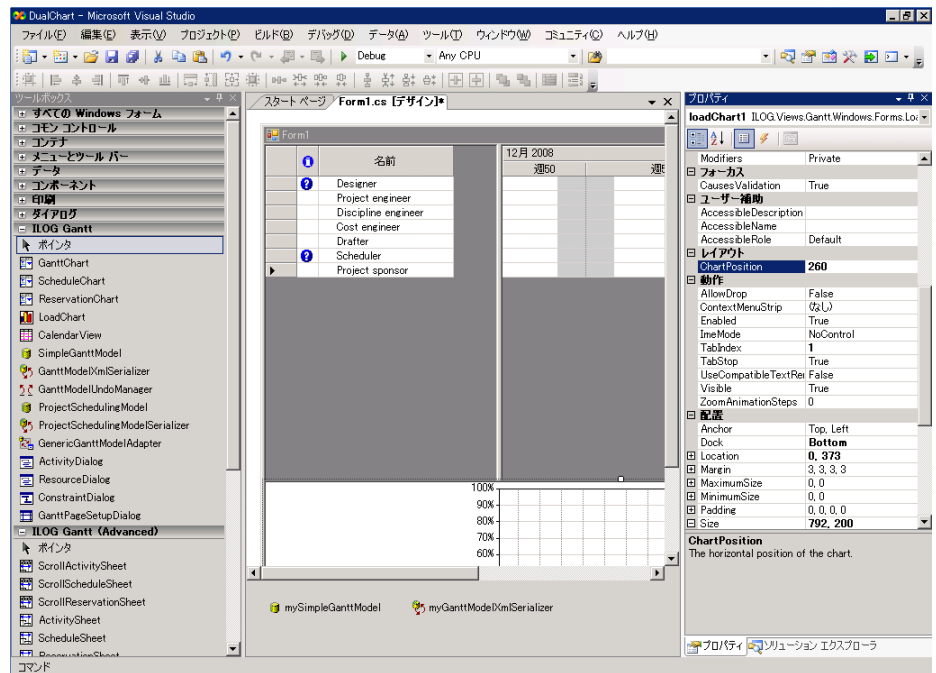


Dock プロパティの隣の矢印をクリックすると、次のコントロールが表示されま  
す。

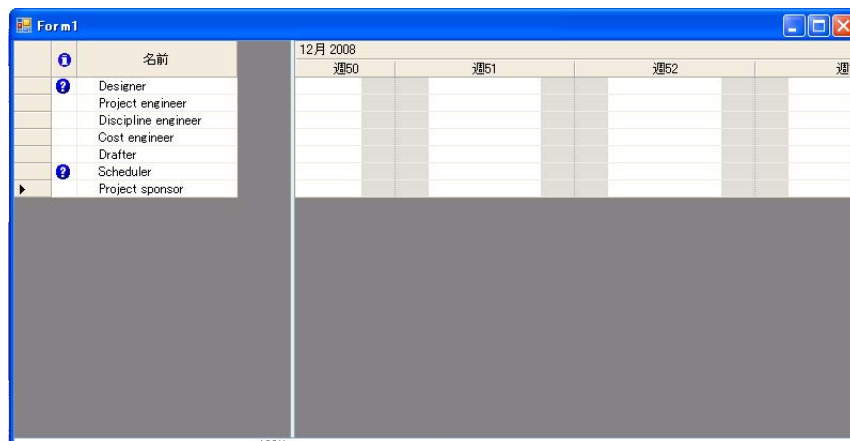


5. 下揃えオプションを適用するエリアを選択します。
6. スケジュール・チャートを右クリックして、[ 前面に移動する ] をクリックしま  
す。
7. 負荷チャートをスケジュール・チャートの区切り線に揃えます。  
Form1 の負荷チャートをクリックします。プロパティ・エクスプローラで  
[ 配置 ] > [ ChartPosition ] とクリックし、**260** と入力します。





これで、負荷チャートが追加されました。アプリケーションは次のように表示されます。



8. 実行したアプリケーションを終了し、Form1.csに戻ります。

## チャートの同期化

次に、作成したスケジュール・チャートと負荷チャートを同期化する方法について説明します。

### 関連項目

[表示時間範囲の同期化](#) | [グリッドの同期化](#) | [チャート区切り線の同期化](#) | [リソース選択の同期化](#)

---

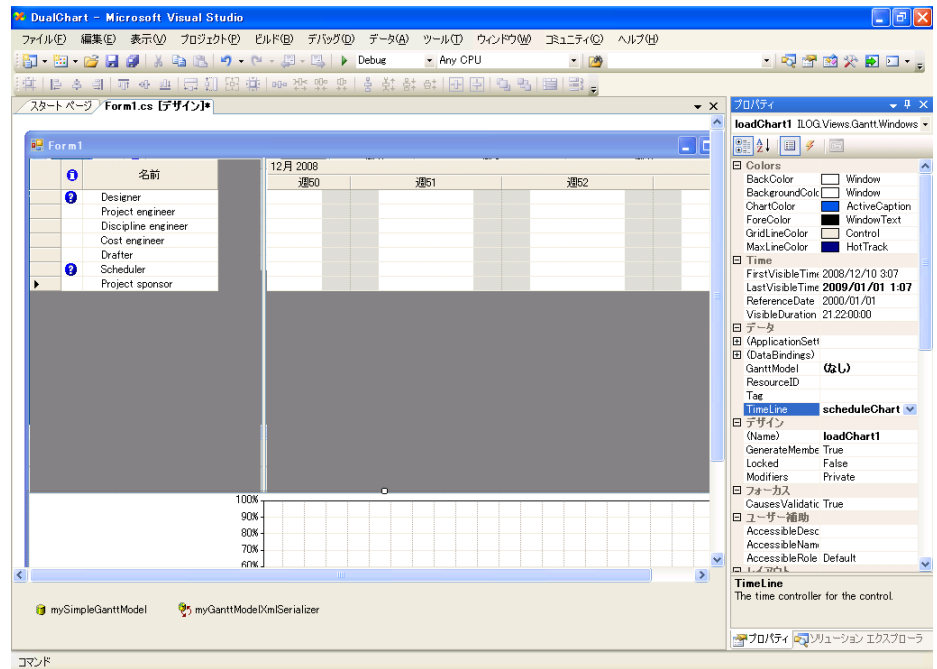
### 表示時間範囲の同期化

時間情報の表示は、**TimeControl** の *time line* (タイム・ライン) というオブジェクトで制御します。このオブジェクトによって、最初に表示される **TimeControl** インスタンスの時間、および時間情報を画面上でピクセルに変更する方法を定義します。

スケジュール・チャートと負荷チャートの同期化は、同じタイム・ラインを共有する場合に行われます。したがって、スケジュールチャートの表示時間を変更すると、同時に負荷チャートにも反映されます。

表示時間範囲を同期化するには、以下の手順を実行します。


1. 負荷チャートをクリックします。
2. プロパティ・エクスプローラの [**TimeLine**] をクリックし、次にドロップダウン・リストの [**scheduleChart**] をクリックします。

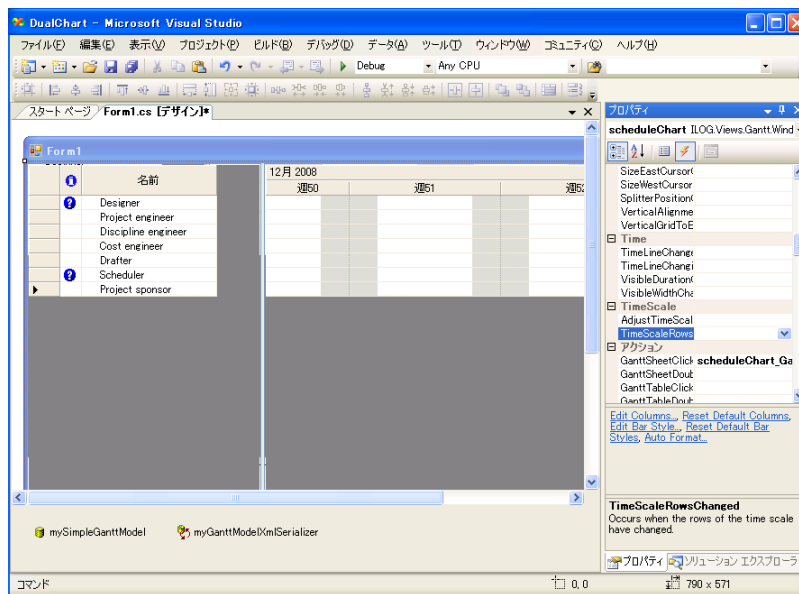


## グリッドの同期化

時間情報を表示する他のコントロール（負荷チャートなど）とスケジュール・チャートを併用する場合は、コントロールの垂直グリッド線（時間グリッド）と時間尺の目盛りを同期化すると便利です。

グリッドを同期化するには、以下の手順を実行します。

1. スケジュール・チャートをクリックします。
2. プロパティ・エクスプローラの  アイコンをクリックし、[イベント] ページを選択します。
3. [TimeScale] > [TimeScaleRowsChanged] イベントをダブルクリックします。



コード・エディタが開き、カーソルが自動的に適切な場所に配置されています。

4. 次のコードを入力します。


```
loadChart.TimeGrids.Synchronize(scheduleChart.TimeGrids);
```

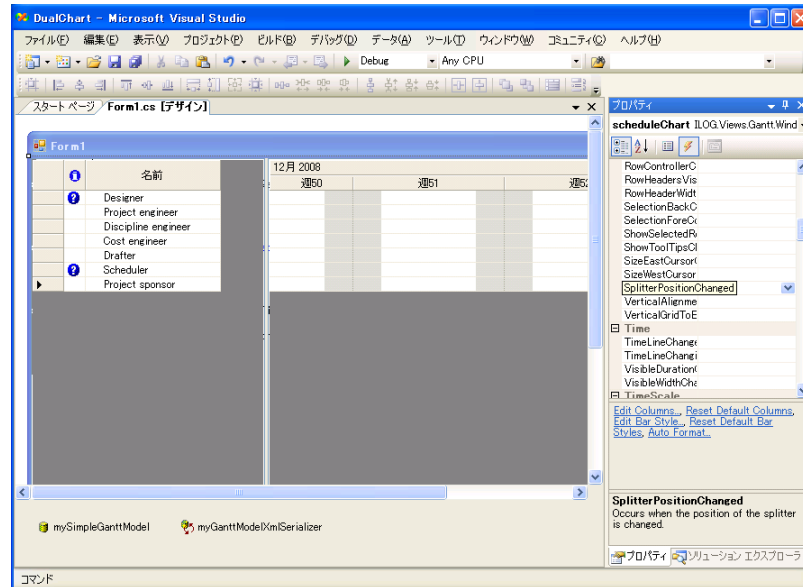
5. ファイル **Form1.cs** を保存し、メニュー・バーの [表示]>[デザイナ] をクリックして、デザイン・モードに戻ります。

### チャート区切り線の同期化

スケジュール・チャートでは、ガント・テーブルとガント・シートが、区切り線で区切られます。区切り線をダブルクリックすると、テーブル全体が表示されます。区切り線は、**SplitterPosition** および **SplitterWidth** プロパティで制御します。

スケジュール・チャートと負荷チャートで、区切り線の位置を同期化するには、以下の手順を実行します。

1. スケジュール・チャートをクリックします。
2. プロパティ・エクスプローラの  アイコンをクリックし、[イベント] ページを選択します。
3. **[SplitterPositionChanged]** イベントをダブルクリックします。



コード・エディタが開き、カーソルが自動的に適切な場所に配置されています。

4. 次のコードを入力します。

```
loadChart.ChartPosition = scheduleChart.SplitterPosition +
scheduleChart.SplitterWidth;
```

5. ファイル **Form1.cs** を保存し、メニュー・バーの [ 表示 ] > [ デザイナ ] をクリックして、デザイン・モードに戻ります。


## リソース選択の同期化

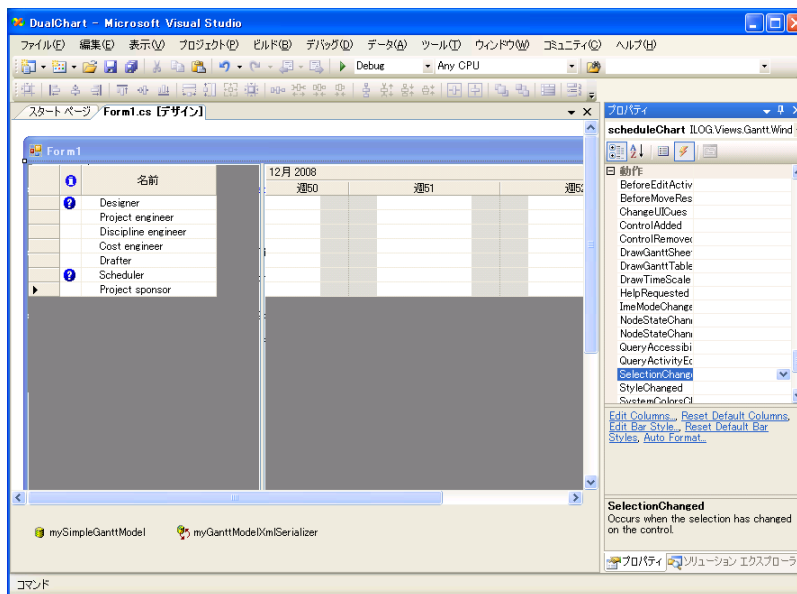
スケジュール・チャートと負荷チャートのリソース選択の同期化は、リソースを選択し、両方のチャートに表示されると行われます。

リソース選択を同期化するには、**Resource** プロパティを使用して、負荷チャートをリソースに関連付けます。リソースに関連付けられた負荷チャートは、ガント・データ・モデルのイベントを受信し、リソースや予約が変更されるたびに、それ自体を更新します。

リソース選択を同期化するには、以下の手順を実行します。

1. スケジュール・チャートをクリックします。

2. プロパティ・エクスプローラの  アイコンをクリックし、[ イベント ] ページを選択します。
3. [ 振る舞い ] > [ SelectionChanged ] イベントをダブルクリックします。



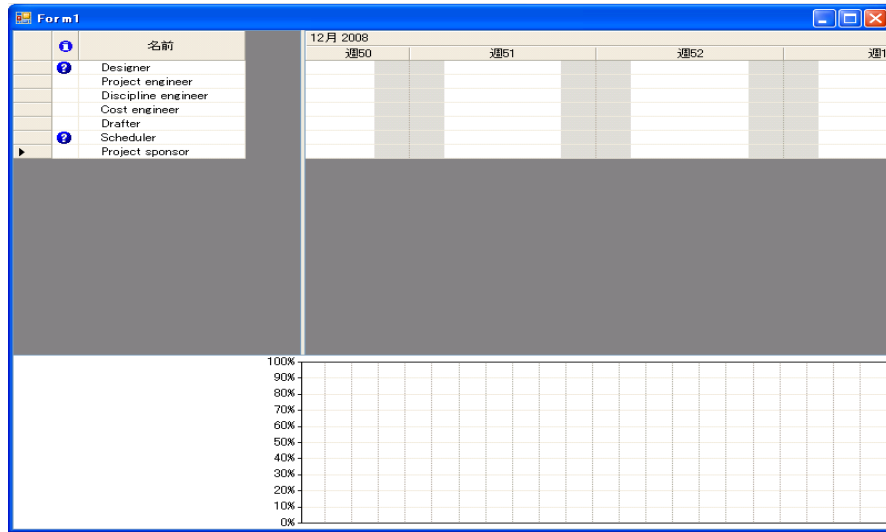
コード・エディタが開き、カーソルが自動的に適切な場所に配置されています。

4. 次のコードを入力します。

```
ILOG.Views.Gantt.Data.IResource[] resources =
scheduleChart.GetSelectedResources();
if (resources.Length == 1)
    loadChart.Resource = resources[0];
else
    loadChart.Resource = null;
```
5. ファイル **Form1.cs** を保存します。
6. コマンド [ デバッグ ] > [ デバッグ開始 ] を使用して、アプリケーションを実行します。

アプリケーション開始時に予約が表示されるようにする

アプリケーションが次のように表示されます。




7. 実行したアプリケーションを終了します。

---

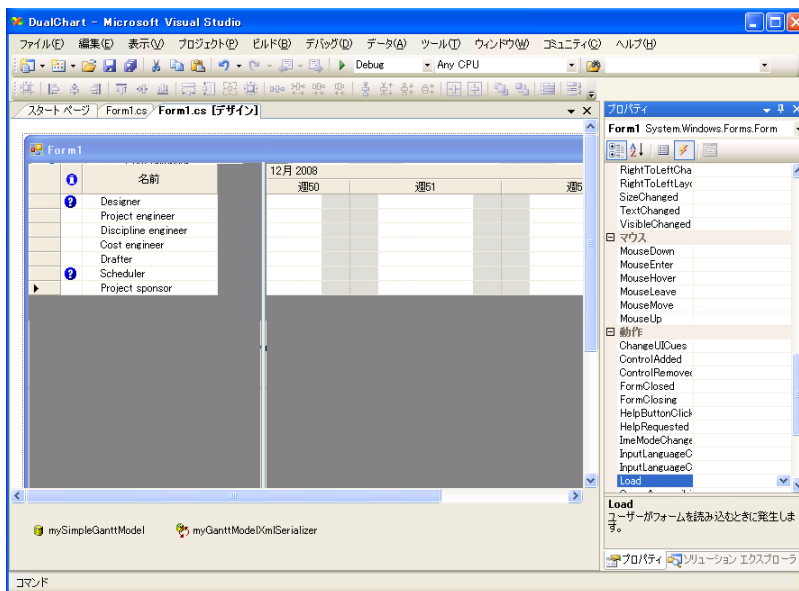
## アプリケーション開始時に予約が表示されるようにする

アプリケーションの開始時は、列や行には何も表示されません。スケジューリング情報は入力されていますが、スクロールしなければ表示されません。

以下のステップでは、アプリケーションの開始時に、チャート内に予約が直ちに表示されるようにする方法について説明します。

1. Form1 を選択します。
2. プロパティ・エクスプローラの  アイコンをクリックし、[ イベント ] ページを選択します。
3. [ 振る舞い ] > [ Load ] イベントをダブルクリックします。

## スケジュール・チャートと負荷チャートの同期化 (Windows Forms チュートリアル)



コード・エディタが開き、カーソルが自動的に適切な場所に配置されています。

4. 次のコードを入力します。

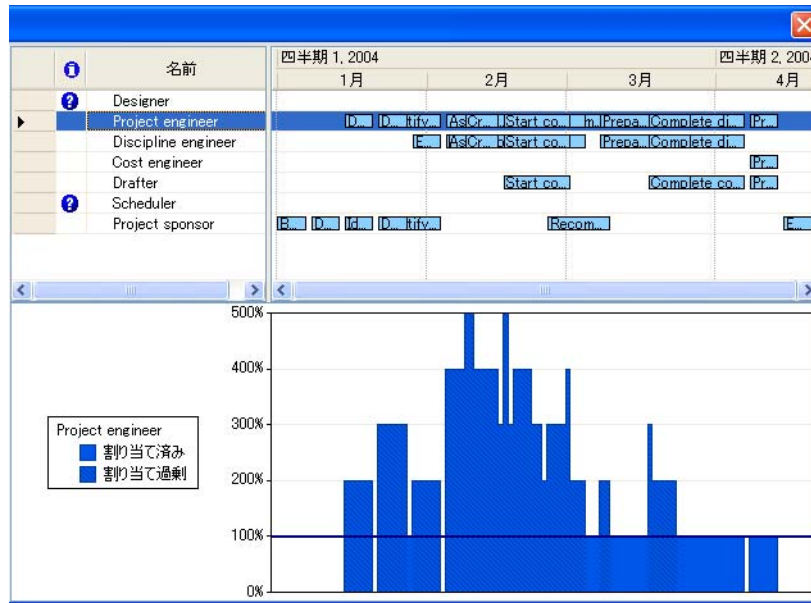
```
scheduleChart.GanttSheet.ZoomToFit(false);
```

5. ファイル **Form1.cs** を保存します。
6. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。



アプリケーション開始時に予約が表示されるようにする

アプリケーションが次のように表示されます。





## カスタム・ガント表示の作成 (Windows Forms チュートリアル)

このチュートリアルでは、アクティビティ・テーブルと2つのアクティビティ・シートから成るカスタム・ガントチャート表示を作成する方法を順番に説明します。最初にテーブルを作成し、スクロール式のアクティビティ・シートを1つずつ追加します。

### 関連項目

[アクティビティ・テーブルの作成 | ガント・モデルとアクティビティ・テーブルの関連付け | アクティビティ・テーブルへのXML データ入力 | スクロール式アクティビティ・シートの追加 | アプリケーション開始時に予約が表示されるようにする](#)

---

### アクティビティ・テーブルの作成

IBM® ILOG® Gantt for .NET では、アクティビティ・テーブルを用いて、スケジューリング情報の表示を制御できます。ユーザーが作成するカスタム表示は、次の3つの部分で構成されます。左側のアクティビティ・テーブル(クラス `GanttTable` のインスタンス)には、データ・モデルからの情報が表示され、右側のスクロール式アクティビティ・シート2つ(クラス `GanttSheet` のインスタンス)には、アクティビティの予約が表示されます。

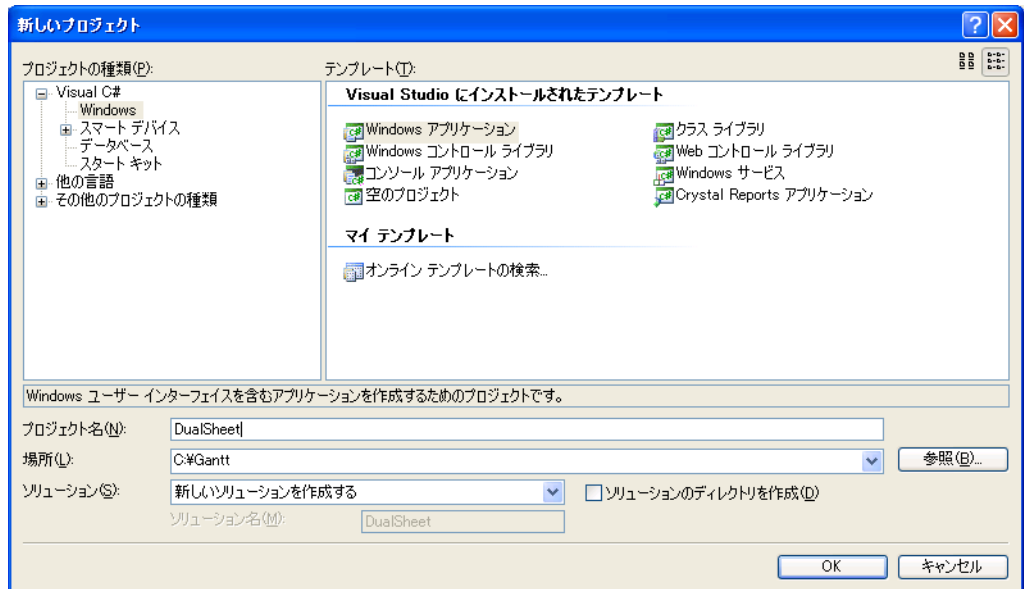
## カスタム・ガント表示の作成 (Windows Forms チュートリアル)

アクティビティ・テーブルを作成するには、以下の手順を実行します。

### 1. WindowsR Forms タイプのプロジェクトを作成します。

[ファイル] メニューの [新規作成] をクリックし、次に [プロジェクト] をクリックします。

[新しいプロジェクト] ウィンドウが表示されます。



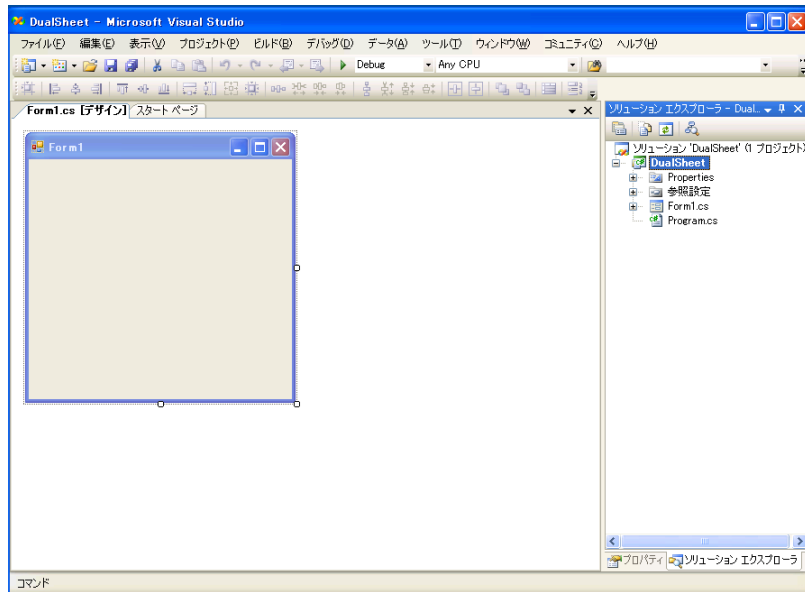
### 2. 左側のウィンドウで、[Visual C#] > [Windows] をクリックします。

### 3. 右側のウィンドウで [Windows アプリケーション] をクリックします。

### 4. プロジェクトに DualSheet という名前を付けます。

### 5. プロジェクトを C:\Gantt に配置します。

Visual StudioR.NET によって、Form1 という名前のフォームが作成され、デフォルトで、ファイル Form1.cs に保存されます。



6. ウィンドウのサイズを変更します。

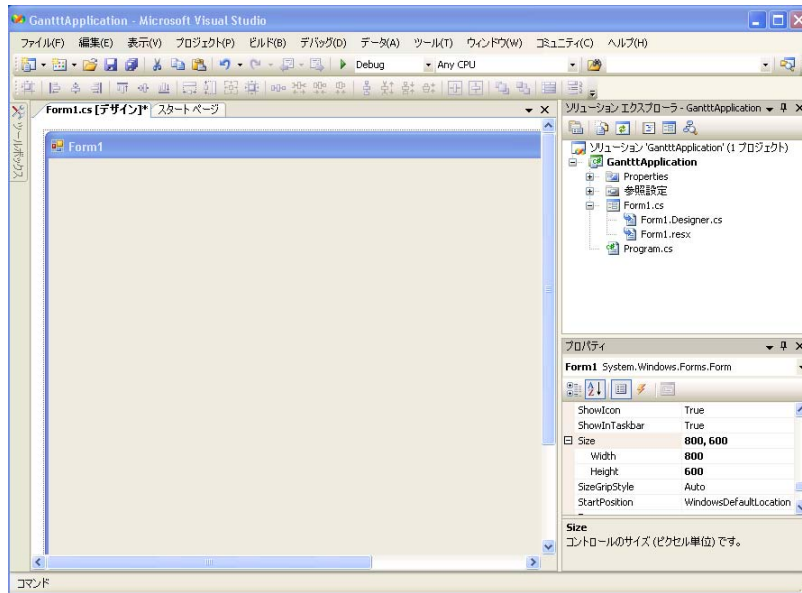
Form1 をクリックし、ウィンドウの右下にある [プロパティ] タブをクリックします。

プロパティ・エクスプローラで [ 配置 ] > [Size] とクリックします。プロパティを展開し、次の値を入力します。

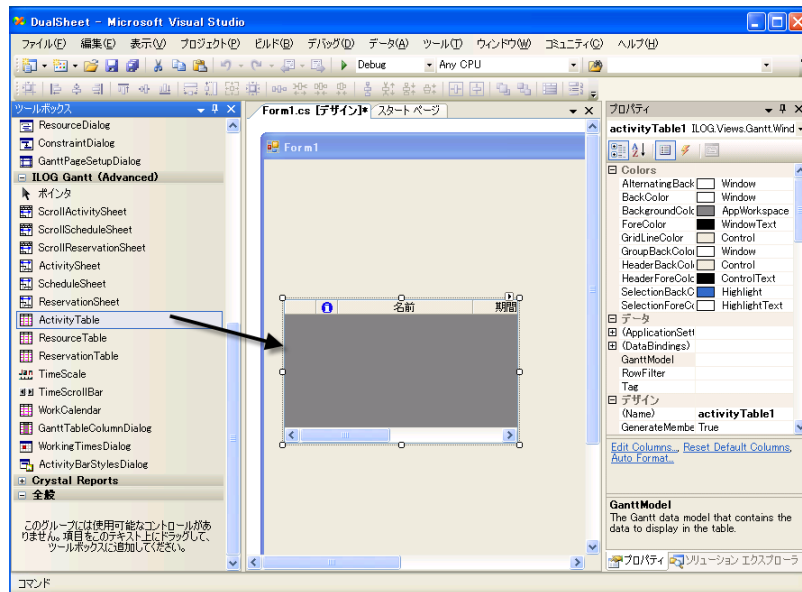
幅 : 800

高さ : 600

## カスタム・ガント表示の作成 (Windows Forms チュートリアル)

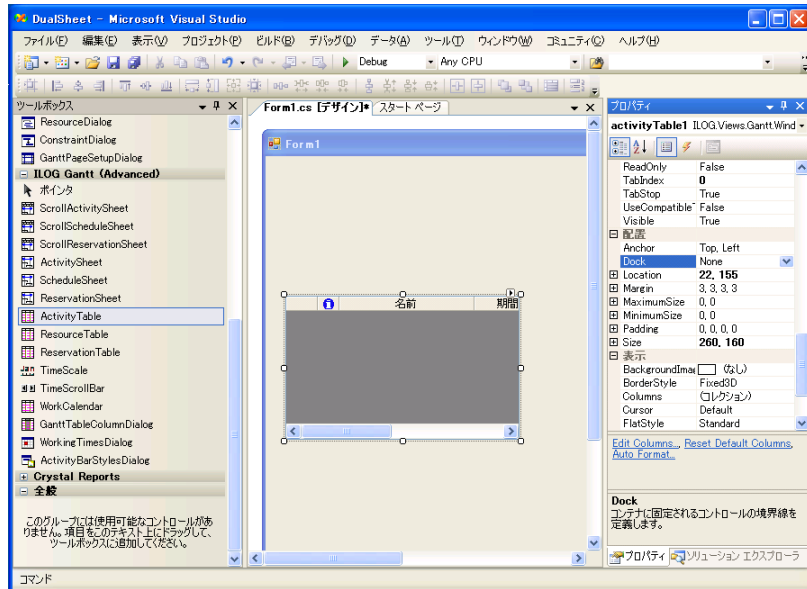


7. IBM ILOG Gantt for .NET ツールボックスで、**ActivityTable** オブジェクトを選択して Form1 までドラッグします。

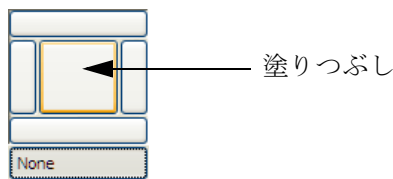


8. アクティビティ・テーブルがフォーム全体に表示されるよう、Dock プロパティを変更します。

アクティビティ・テーブルをクリックし、プロパティ・エクスプローラで [ 配置 ] > [ Dock ] とクリックします。



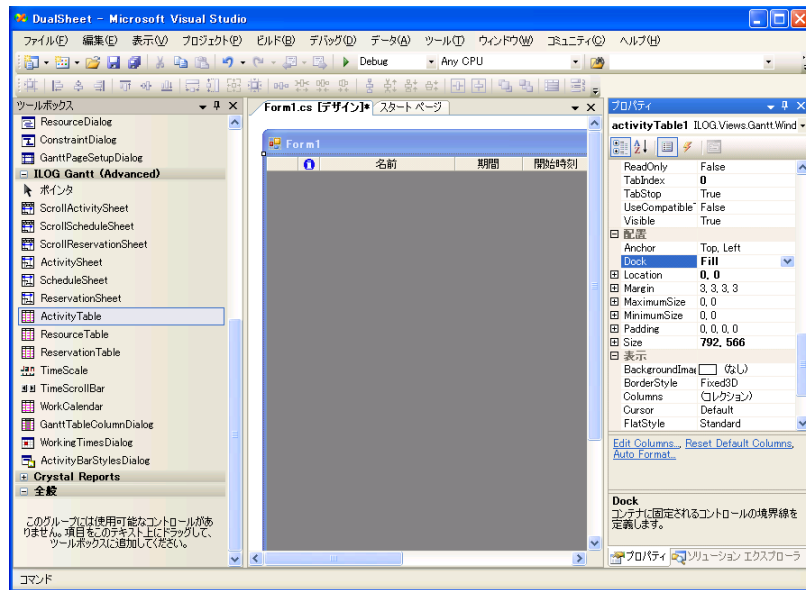
Dock プロパティの隣の矢印をクリックすると、次のコントロールが表示されます。



9. 塗りつぶしオプションを適用するエリアを選択します。

Form1 は次のように表示されます。

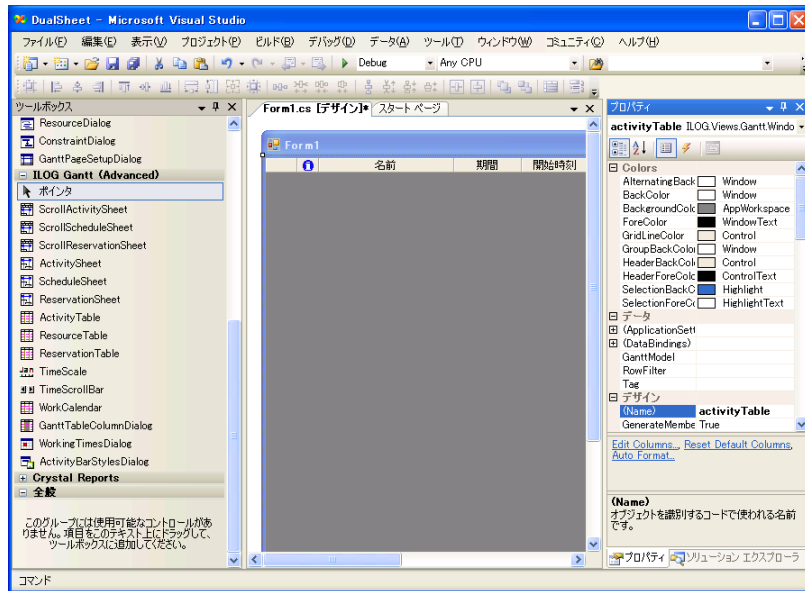
## カスタム・ガント表示の作成 (Windows Forms チュートリアル)



### 10. アクティビティ・テーブルの名前を変更します。

Form1 でアクティビティ・テーブルをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**activityTable** という新しい名前を入力します。



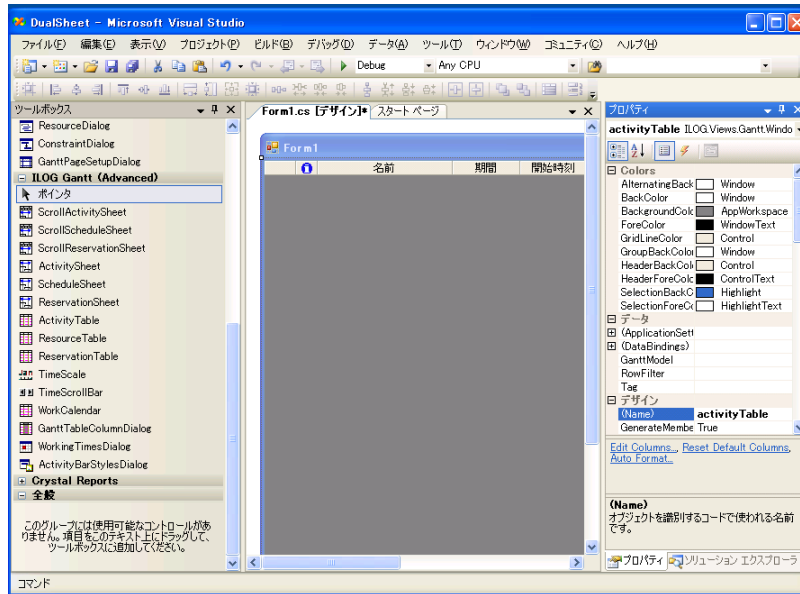


アクティビティ・テーブルの外観は、ユーザーのニーズに合わせて、完全にカスタマイズできます。

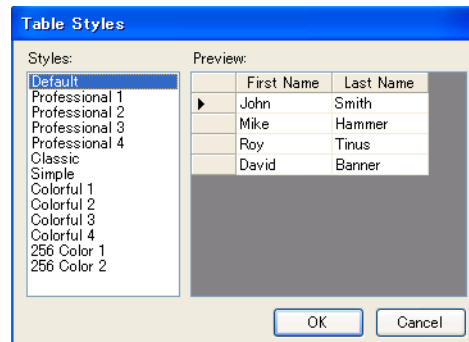
11. 作成したアクティビティ・テーブルのスタイルを定義します。

Form1 のアクティビティ・テーブルを右クリックし、ショートカット・メニューの [AutoFormat] をクリックするか、プロパティ・エクスプローラの下部に表示される [AutoFormat] をクリックします。

## カスタム・ガント表示の作成 (Windows Forms チュートリアル)



[テーブル・スタイル] ウィンドウが表示されます。



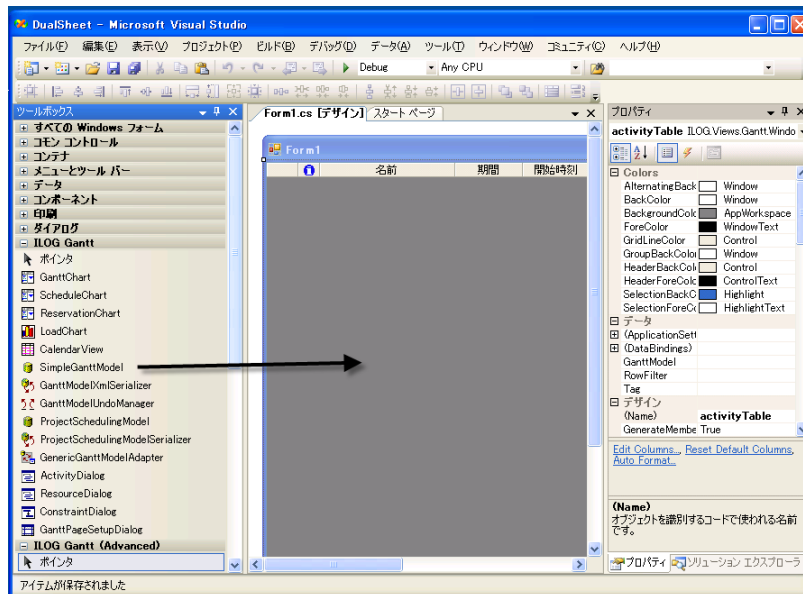
スタイルを選択し、[OK] をクリックします。選択したスタイルが、アクティビティ・テーブルに自動的に反映されます。

12. ファイル **Form1.cs** を保存します。

## ガント・モデルとアクティビティ・テーブルの関連付け

ガント・データ・モデルには、アクティビティ・テーブルに表示されるスケジュール情報が含まれます。以下では、簡単なガント・データ・モデルをアクティビティ・テーブルに結合する方法について説明します。

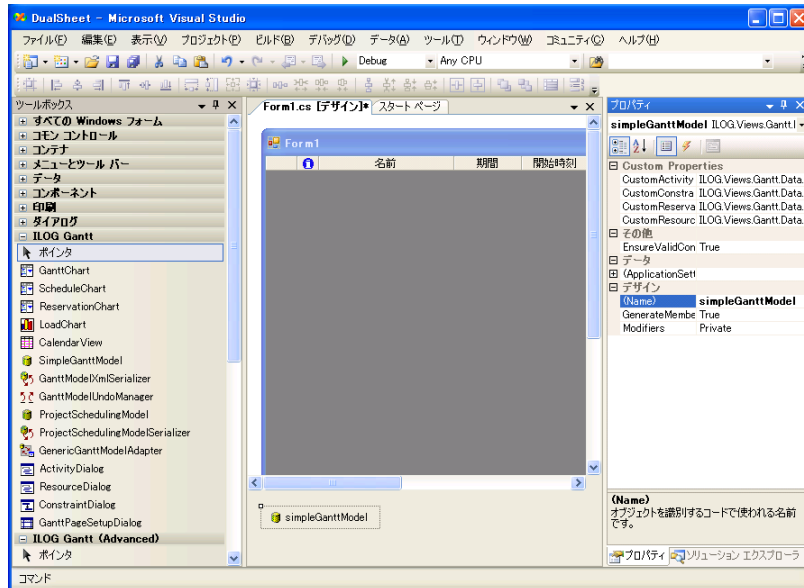
1. IBM® ILOG® Gantt for .NET ツールボックスで、**SimpleGanttModel** オブジェクトを選択して Form1 までドラッグします。



2. この簡単なガント・モデルの名前を変更します。

Form1 の [**SimpleGanttModel1**] オブジェクトをクリックします。プロパティ・エクスプローラで [**デザイン**] > [**Name**] とクリックし、**simpleGanttModel** という新しい名前を入力します。

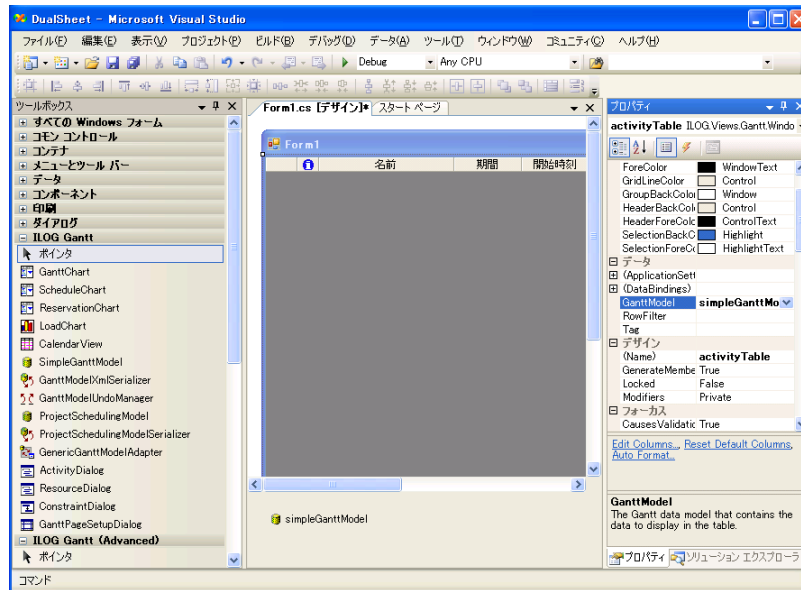
## カスタム・ガント表示の作成 (Windows Forms チュートリアル)



3. ガント・モデルとアクティビティ・テーブルを関連付けます。

Form1 のアクティビティ テーブルをクリックし、プロパティ・エクスプローラから [ データ ] > [GanttModel] とクリックします。

GanttModel プロパティの横に表示されるドロップダウン・リストの矢印をクリックし、[simpleGanttModel] をクリックします。



これで、簡単なガント・モデルが Form1 に表示されるアクティビティ・テーブルに関連付けられました。

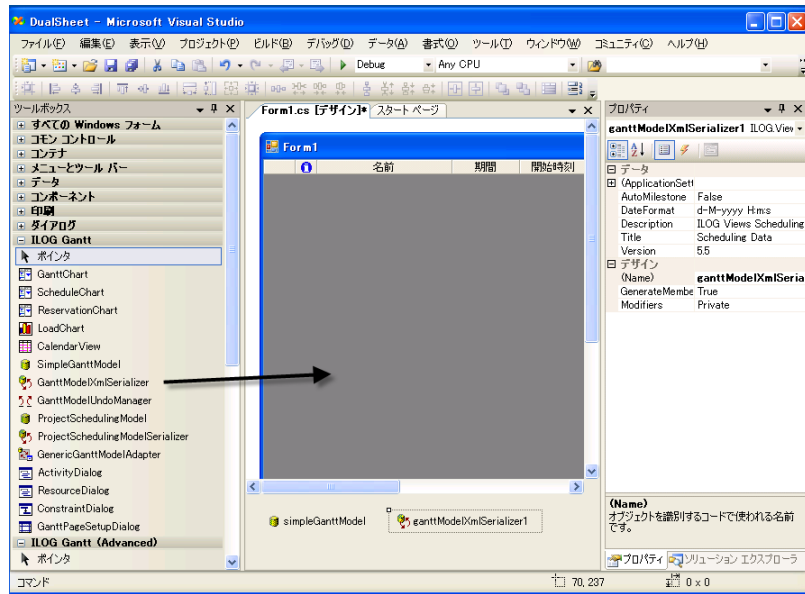
## アクティビティ・テーブルへの XML データ入力

IBM® ILOG® Gantt for .NET では、Scheduling Data Exchange Language (SDXL) ファイルのスケジューリング・データをシリアル化・非シリアル化できます。これには、**GanttModelXmlSerializer** クラスを使用します。

SDXL ファイルからのスケジューリング・データを非シリアル化した後、この情報をアクティビティ・テーブルへの入力に使用するには、以下の手順を実行します。

1. IBM ILOG Gantt for .NET ツールボックスで、**GanttModelXmlSerializer** オブジェクトを選択して Form1 までドラッグします。

## カスタム・ガント表示の作成 (Windows Forms チュートリアル)

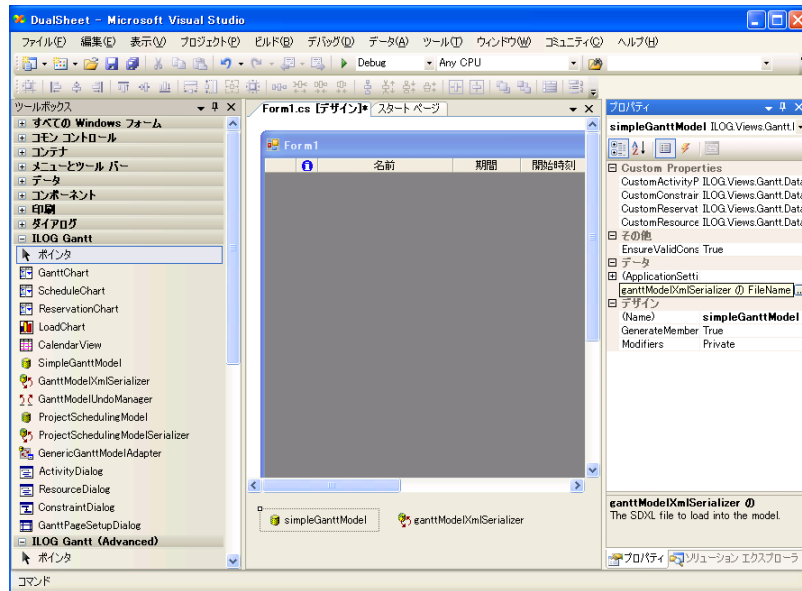


2. このガント・モデル・シリアライザの名前を変更します。

Form1 の [ganttModelXmlSerializer1] オブジェクトをクリックします。プロパティ・エクスプローラで [デザイン] > [Name] とクリックし、**ganttModelXmlSerializer** という新しい名前を入力します。

3. **simpleGanttModel** に戻り、クリックします。

**ganttModelXmlSerializer** の **FileName** という名前の新規プロパティが作成されています。

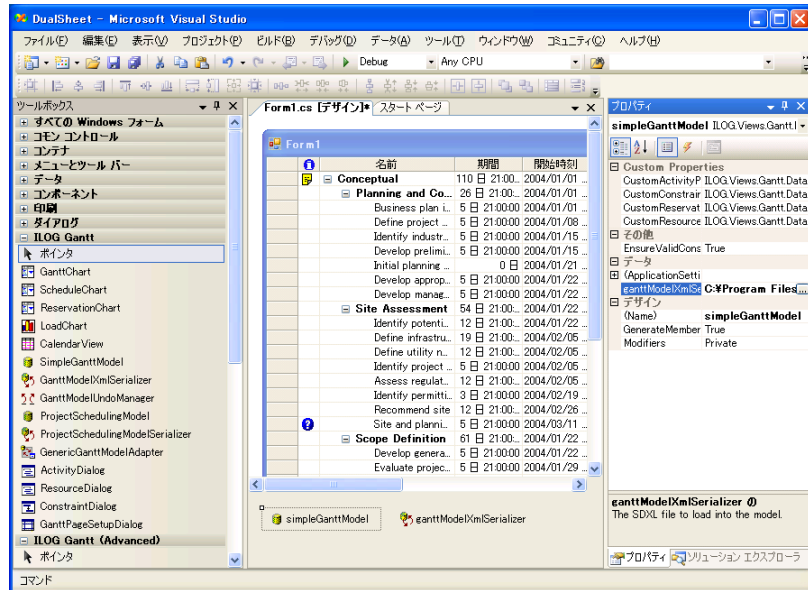


4. プロパティ [ganttModelXmlSerializer] の [FileName] をクリックし、隣の3つの点が表示されているボタンをクリックします。アクティビティ・テーブルにデータを入力する SDXL ファイルをクリックします。

**注:** サンプル SDXL ファイルは <install-dir>\Data にあります。

SDXL ファイルに含まれるデータがアクティビティ・テーブルに表示されます。

## カスタム・ガント表示の作成 (Windows Forms チュートリアル)



5. ファイル **Form1.cs** を保存します。
6. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

この段階で、アクティビティ・テーブルには、テーブルが 1 つだけ含まれており、次のように表示されます。

名前	期間	開始時刻	終了時刻	リソース
Conceptual	110 日	21:00.. 2004/01/01 ..	2004/04/21 ..	
Planning and Co...	26 日	21:00.. 2004/01/01 ..	2004/01/28 ..	
Business plan i...	5 日	21:00:00 2004/01/01 ..	2004/01/07 ..	Project sponsor
Define project ...	5 日	21:00:00 2004/01/08 ..	2004/01/14 ..	Project sponsor
Identify industr...	5 日	21:00:00 2004/01/15 ..	2004/01/21 ..	Project sponsor,P...
Develop prelimi...	5 日	21:00:00 2004/01/15 ..	2004/01/21 ..	Project engineer
Initial planning ...	0 日	2004/01/21 ..	2004/01/21 ..	
Develop approp...	5 日	21:00:00 2004/01/22 ..	2004/01/28 ..	Project sponsor
Develop manag...	5 日	21:00:00 2004/01/22 ..	2004/01/28 ..	Project sponsor,P...
Site Assessment	54 日	21:00.. 2004/01/22 ..	2004/03/17 ..	
Identify potenti...	12 日	21:00.. 2004/01/22 ..	2004/02/04 ..	Project sponsor,P...
Define infrastru...	19 日	21:00.. 2004/02/05 ..	2004/02/25 ..	Project engineer
Define utility n...	12 日	21:00.. 2004/02/05 ..	2004/02/18 ..	Project engineer...
Identify project ...	5 日	21:00:00 2004/02/05 ..	2004/02/11 ..	Project engineer...
Assess regulat...	12 日	21:00.. 2004/02/05 ..	2004/02/18 ..	Project engineer...
Identify permitt...	3 日	21:00:00 2004/02/19 ..	2004/02/23 ..	Project engineer
Recommend site	12 日	21:00.. 2004/02/26 ..	2004/03/10 ..	Project sponsor,P...
Site and planni...	5 日	21:00:00 2004/03/11 ..	2004/03/17 ..	
Scope Definition	61 日	21:00.. 2004/01/22 ..	2004/03/24 ..	
Develop genera...	5 日	21:00:00 2004/01/22 ..	2004/01/28 ..	Project sponsor,P...
Evaluate projec...	5 日	21:00:00 2004/01/29 ..	2004/01/29 ..	Project sponsor,P...

7. 実行したアプリケーションを終了します。

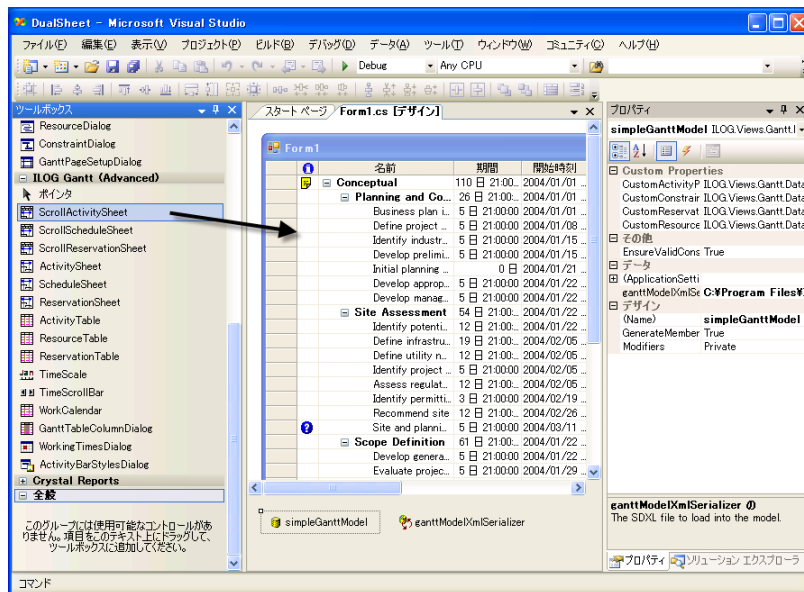


## スクロール式アクティビティ・シートの追加

次に、スクロール式アクティビティ・シートと、作成したテーブルを関連付け同期化する方法について説明します。

最初のスクロール式アクティビティ・シートの追加

1. IBM® ILOG® Gantt for .NET ツールボックスで、**ScrollActivitySheet** オブジェクトを選択して Form1 までドラッグします。



2. スクロール式アクティビティ・シートのサイズを変更します。

Form1 のスクロール式アクティビティ・シートをクリックします。プロパティ・エクスプローラで [ 配置 ] > [Size] とクリックします。プロパティを展開し、次の値を入力します。

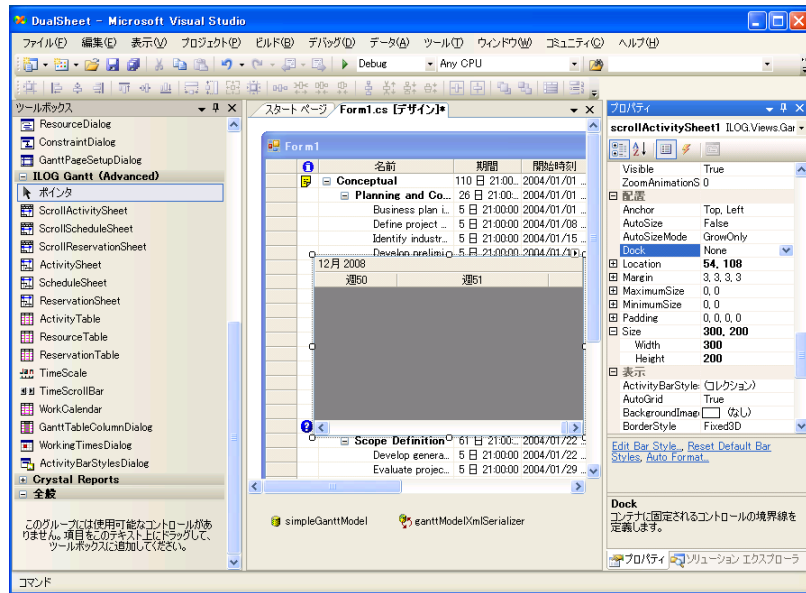
幅 : 300

高さ : デフォルトのまま

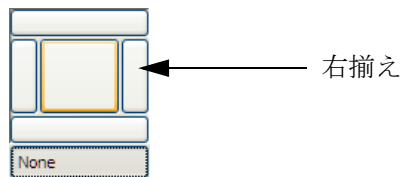
3. スクロール式アクティビティ・シートがアクティビティ・テーブルの右側に揃うように、Dock プロパティを変更します。

Form1 の [scrollActivitySheet1] オブジェクトをクリックし、プロパティ・エクスプローラで [ 配置 ] > [Dock] とクリックします。

## カスタム・ガント表示の作成 (Windows Forms チュートリアル)

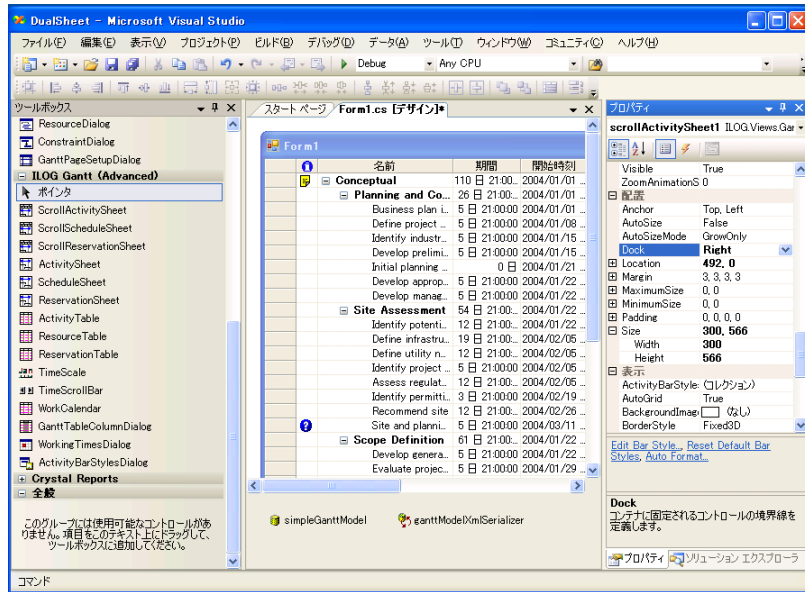


Dock プロパティの隣の矢印をクリックすると、次のコントロールが表示されます。



4. 右揃えオプションを適用するエリアを選択します。

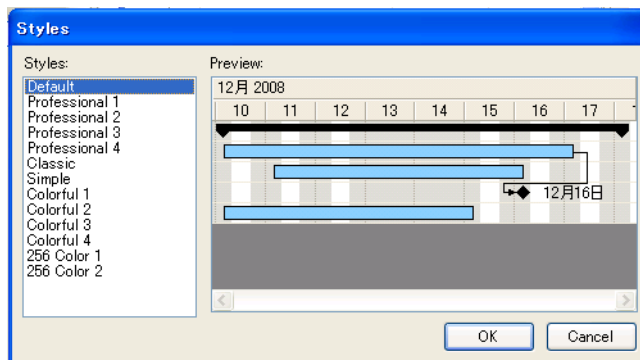
これで、スクロール式アクティビティ・シートがアクティビティ・テーブルの右側に揃いました。必要に応じて、水平方向にスクロールして表示します。



- 作成したスクロール式アクティビティ・シートのスタイルを定義します。

Form1 のスクロール式アクティビティ・シートを右クリックし、コンテキストメニューの [AutoFormat] をクリックするか、プロパティ・エクスプローラの下部に表示される [AutoFormat] をクリックします。

[スタイル] ウィンドウが表示されます。

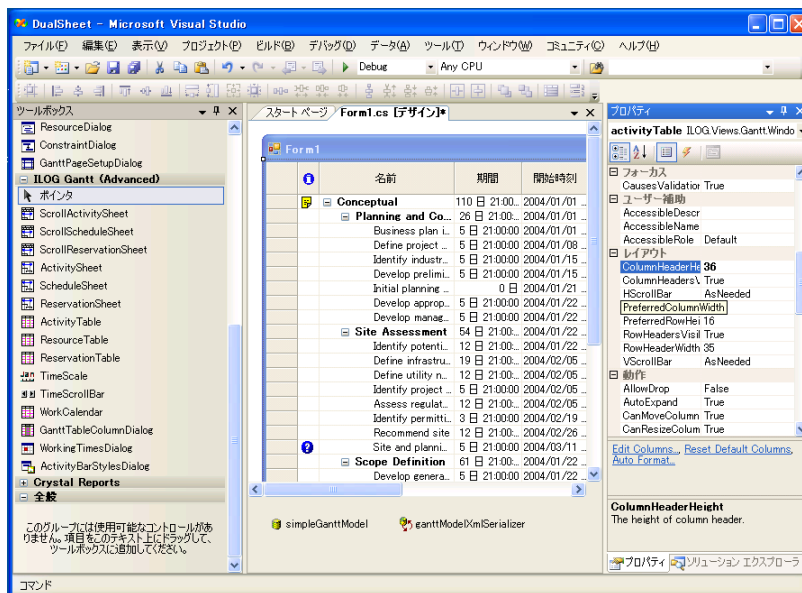


スタイルを選択し、[OK] をクリックします。選択したスタイルが、スクロール式アクティビティ・シートに自動的に反映されます。

- アクティビティ・テーブルに戻り、右クリックして、[前面に移動する] をクリックします。

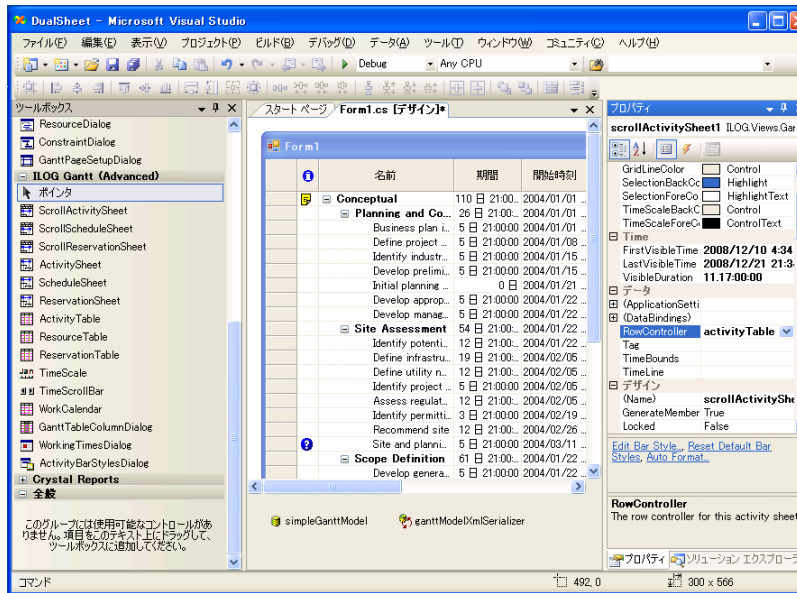
7. テーブル・ヘッダーと時間尺を同期化します。

アクティビティ・テーブルをクリックし、次にプロパティ・エクスプローラで [ 配置 ] > [ColumnHeaderHeight] とクリックして、36 と入力します。



8. アクティビティ・テーブルとスクロール式アクティビティ・シートを垂直方向に同期化します。

スクロール式アクティビティ・シートをクリックし、次にプロパティ・エクスプローラで [ データ ] > [RowController] とクリックします。[RowController] プロパティの横に表示されるドロップダウンの矢印をクリックし、[activityTable] をクリックします。



9. ファイル Form1.cs. を保存します。

10. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

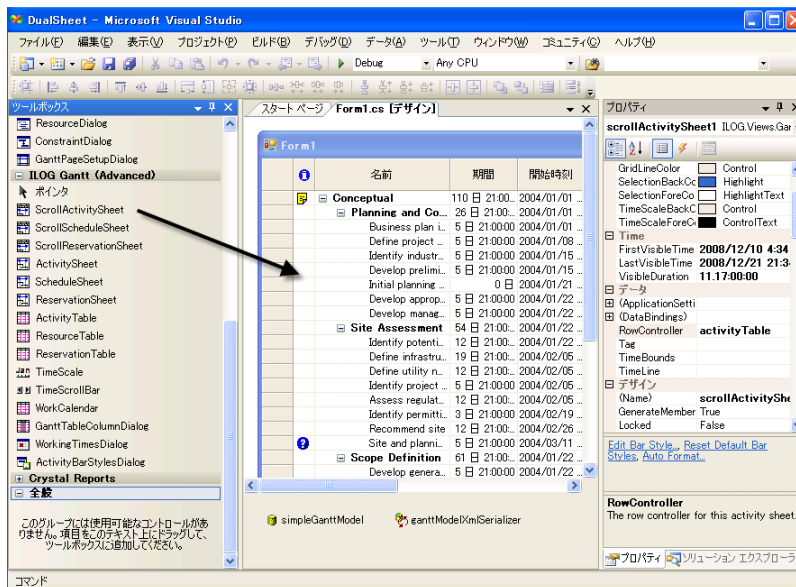
この段階で、ガント・チャート・アプリケーションには、テーブルとシートが1つずつ含まれており、次のように表示されます。

名前	期間	開始時刻	終了時刻	リソ
Conceptual	110 日 21:00.. 2004/01/01 .. 2004/04/21 ..			12月 2008 週50
Planning and Co...	26 日 21:00.. 2004/01/01 .. 2004/01/28 ..			週51
Business plan i...	5 日 21:00:00 2004/01/01 .. 2004/01/07 .. Project spd			
Define project ...	5 日 21:00:00 2004/01/08 .. 2004/01/14 .. Project spd			
Identify industr...	5 日 21:00:00 2004/01/15 .. 2004/01/21 .. Project spd			
Develop prelimi...	5 日 21:00:00 2004/01/15 .. 2004/01/21 .. Project ene			
Initial plannin...	0 日 2004/01/21 .. 2004/01/21 ..			
Develop approv...	5 日 21:00:00 2004/01/22 .. 2004/01/28 .. Project spd			
Develop manag...	5 日 21:00:00 2004/01/22 .. 2004/01/28 .. Project spd			
Site Assessment	54 日 21:00.. 2004/01/22 .. 2004/03/17 ..			
Identify potenti...	12 日 21:00.. 2004/01/22 .. 2004/02/04 .. Project spd			
Define infrastru...	19 日 21:00.. 2004/02/05 .. 2004/02/25 .. Project ene			
Define utility n...	12 日 21:00.. 2004/02/05 .. 2004/02/11 .. Project ene			
Identify project ...	5 日 21:00:00 2004/02/05 .. 2004/02/11 .. Project ene			
Assess regulat...	12 日 21:00.. 2004/02/05 .. 2004/02/18 .. Project ene			
Identify permitti...	3 日 21:00:00 2004/02/19 .. 2004/02/23 .. Project ene			
Recommend site	12 日 21:00.. 2004/02/26 .. 2004/03/10 .. Project spd			
Site and plann...	5 日 21:00:00 2004/03/11 .. 2004/03/17 ..			
Scope Definition	61 日 21:00.. 2004/01/22 .. 2004/03/17 ..			
Develop genera...	5 日 21:00:00 2004/01/22 .. 2004/01/28 .. Project spd			
Evaluate projec...	5 日 21:00:00 2004/01/29 .. 2004/02/04 .. Project ene			
Start major stu...	0 日 21:00:00 2004/02/05 .. 2004/02/06 .. Discipline i			
Complete major...	17 日 21:00.. 2004/02/09 .. 2004/02/27 .. Discipline i			
Develop specifi...	3 日 21:00:00 2004/03/01 .. 2004/03/05 .. Project ene			
Prepare final co...	10 日 21:00.. 2004/03/08 .. 2004/03/19 .. Project ene			
Provide written ...	5 日 21:00:00 2004/03/18 .. 2004/03/24 .. Project ene			
Discipline Supp...	57 日 21:00.. 2004/02/09 .. 2004/04/07 ..			
Create delivera...	6 日 21:00:00 2004/02/09 .. 2004/02/16 .. Project ene			
Start flow shee...	5 日 21:00:00 2004/02/17 .. 2004/02/23 .. Discipline i			
Start disciplin...	13 日 21:00.. 2004/02/17 .. 2004/03/02 .. Project ene			
Start conceptua...	13 日 21:00.. 2004/02/17 .. 2004/03/02 .. Discipline i			
Complete flow ...	19 日 21:00.. 2004/03/18 .. 2004/04/07 .. Discipline i			

11. 実行したアプリケーションを終了します。

## 2 番目のスクロール式アクティビティ・シートの追加

1. IBM ILOG Gantt for .NET ツールボックスで、**ScrollActivitySheet** オブジェクトを選択して Form1 までドラッグします。



2. 2 番目のスクロール式アクティビティ・シートのサイズを変更します。

Form1 のスクロール式アクティビティ・シートをクリックします。プロパティ・エクスプローラで [ 配置 ] > [Size] とクリックし、プロパティを展開して、次の値を入力します。

幅 : 300

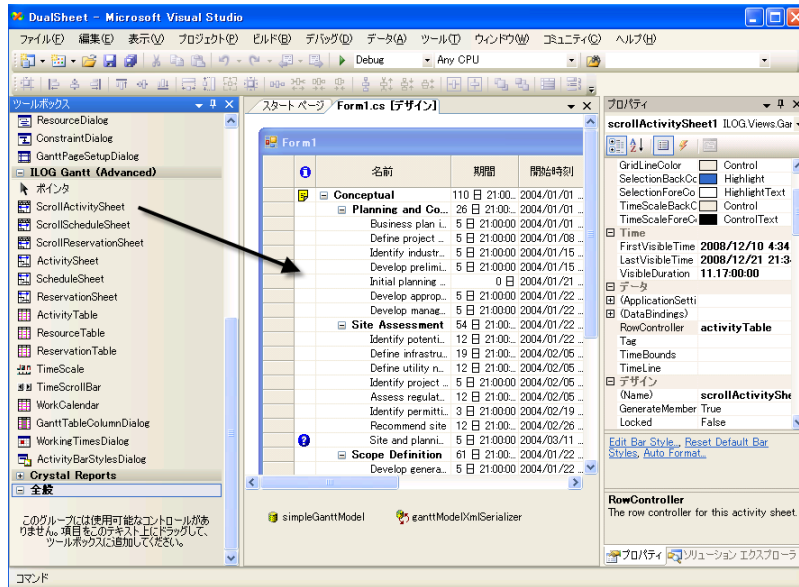
高さ : デフォルトのまま

3. スクロール式アクティビティ・シートがアクティビティ・テーブルの右側に揃うように、Dock プロパティを変更します。

Form1 の 2 番目のスクロール式アクティビティ・シートをクリックし、プロパティ・エクスプローラで [ 配置 ] > [Dock] とクリックします。

4. 右揃えオプションを適用するエリアを選択します。

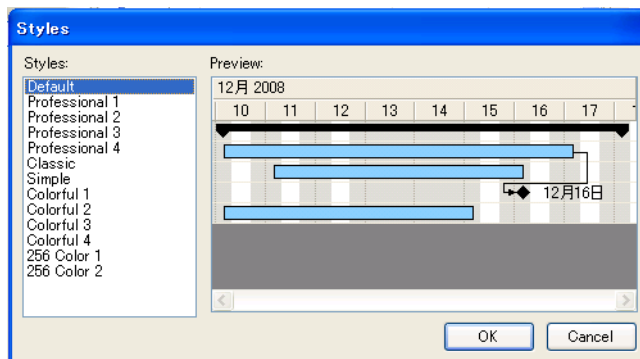
これで、2 番目のスクロール式アクティビティ・シートがアクティビティ・テーブルの右側に揃いました。



- 作成したスクロール式アクティビティ・シートのスタイルを定義します。

Form1 のスクロール式アクティビティ・シートを右クリックし、コンテキストメニューの [AutoFormat] をクリックするか、プロパティ・エクスプローラの下部に表示される [AutoFormat] をクリックします。

[スタイル] ウィンドウが表示されます。



スタイルを選択し、[OK] をクリックします。選択したスタイルが、スクロール式アクティビティ・シートに自動的に反映されます。

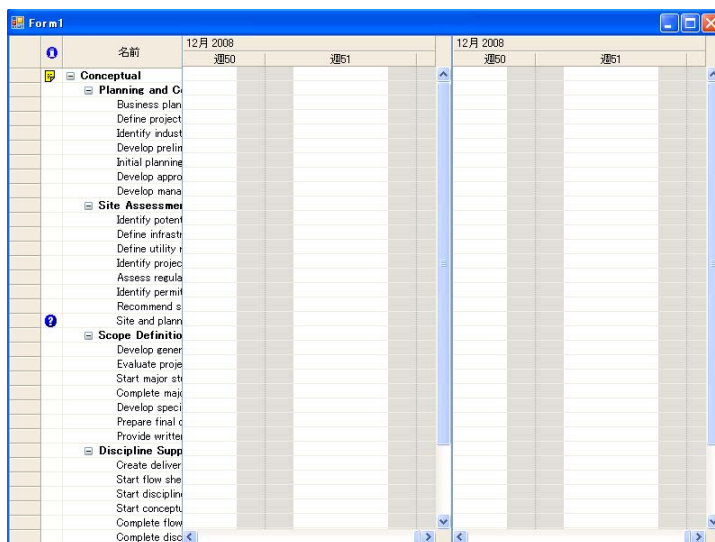
- アクティビティ・テーブルに戻り、右クリックして、[前面に移動する] をクリックします。

7. アクティビティ・テーブルと 2 番目のスクロール式アクティビティ・シートを垂直方向に同期化します。

2 番目のスクロール式アクティビティ・シートをクリックし、次にプロパティ・エクスプローラで [ データ ] > [RowController] とクリックします。[RowController] プロパティの横に表示されるドロップダウンの矢印をクリックし、[activityTable] をクリックします。

8. ファイル **Form1.cs** を保存します。
9. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

これで、2 番目のスクロール式アクティビティ・シートが追加されました。アプリケーションは次のように表示されます。



10. 実行したアプリケーションを終了します。

---

## アプリケーション開始時に予約が表示されるようにする

読み込む SDXL ファイルによっては、デフォルトの表示時間間隔にデータが入力されていないため、アプリケーションの開始時、スクロール式アクティビティ・シートに何も表示されない場合があります。スケジューリング情報は入力されていますが、スクロールしなければ表示されません。以下のステップでは、アプリケーションの開始時に、予約が直ちに表示されるようにする方法について説明します。最初のスクロール式アクティビティ・シートにスケジューリング情報の最初の半



分が表示され、残りの半分は2番目のスクロール式アクティビティ・シートに表示されます。

1. Design モードで、Form1 のヘッダーをダブルクリックします。

コード・エディタが開き、カーソルが自動的に適切な場所に配置されています。

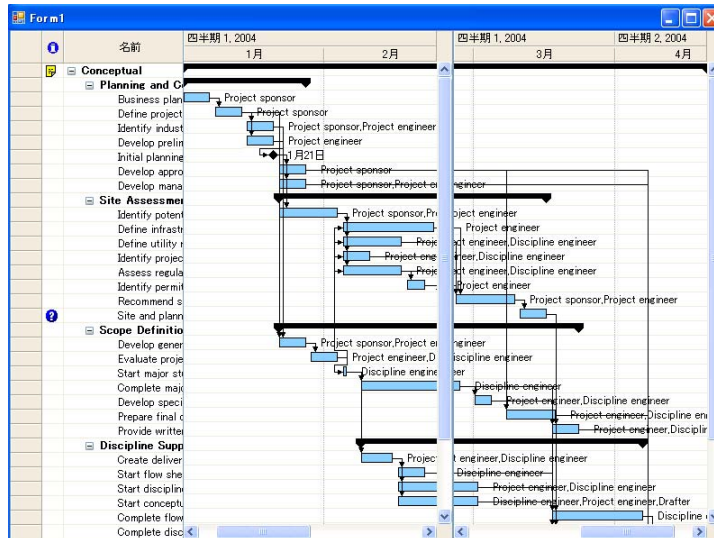
2. 次のコードを入力します。

```
ILOG.Views.Gantt.DateTimeInterval interval =  
scrollActivitySheet1.TimeBounds.TimeInterval;  
scrollActivitySheet2.VisibleTimeInterval = new  
ILOG.Views.Gantt.DateTimeInterval(interval.Start,  
TimeSpan.FromTicks((long)(interval.Duration.Ticks / 2)));  
scrollActivitySheet1.VisibleTimeInterval = new  
ILOG.Views.Gantt.DateTimeInterval(scrollActivitySheet2.LastVisibleTime,  
TimeSpan.FromTicks((long)(interval.Duration.Ticks / 2)));
```

3. ファイル **Form1.cs** を保存します。

4. コマンド [デバッグ] > [デバッグ開始] を使用して、アプリケーションを実行します。

これで、下図のように、アプリケーション開始時にテーブル内に予約が直ちに表





<b>S</b>	
SDXL <b>14, 23</b>	
<b>あ</b>	
アクティビティ・テーブル <b>57</b>	
<b>か</b>	
ガント・シート <b>5, 19</b>	
ガントチャート <b>5, 19</b>	
<b>く</b>	
区切り線 <b>50</b>	
<b>す</b>	
スケジュール・チャート <b>29</b>	
<b>た</b>	
タイム・ライン <b>48</b>	
<b>ち</b>	
チュートリアル	
アクティビティ・テーブルの作成 <b>57</b>	
アプリケーション開始時に予約が表示されるようにする <b>53, 78</b>	
ガントチャートの作成 <b>5, 19</b>	
	ガント・データ・モデルとガントチャートの関連付け <b>13, 23</b>
	ガント・データ・モデルとスケジュール・チャートの関連付け <b>38</b>
	ガント・モデルとアクティビティ・テーブルの関連付け <b>65</b>
	スクロール式アクティビティ・シートの追加 <b>71</b>
	スケジュール・チャートの作成 <b>29</b>
	チャート区切り線の同期化 <b>50</b>
	チャートの同期化 <b>48</b>
	表示時間範囲の同期化 <b>48</b>
	負荷チャートの追加 <b>44</b>
	リソース選択の同期化 <b>51</b>
	<b>ふ</b>
	負荷チャート <b>44</b>
	<b>り</b>
	リソース <b>51</b>

