XRT Widgets 対応 MVC Kit

プログラマーズ・ガイド&リファレンス・マニュアル

UNIX Edition ■ Version 2.0



株式会社タンジェント・コンピューティング
〒150-0001 東京都渋谷区神宮前 6-6-11
tel:(03)3409-5364
fax: (03)3409-0147
info@tan.co.jp
http://www.tan.co.jp

1999 年 5 月

本書は著作権法により保護されています。(株)タンジェント・コンピューティングによる事前の許可なく、本書のいかなる部分も無断転載、複製、複写を禁じます。
Copyright©1999 (株)タンジェント・コンピューティング

XRT、XRT/3d、XRT/care、XRT/field、XRT/gear、XRT/graph、XRT/table、XRT PDSはKL Group Inc.の登録商標です。
その他製品名、名称、サービス等は該当する会社または組織の商標または登録商標です。

目次

Preface 1

MVC について 1
前提知識 1
マニュアルの概要 1

Part I　MVC Kit の使用法

1　簡単なプログラムの開発 5

1.1　はじめに 5
1.2　環境変数 5
1.3　プログラムのコンパイルと許可 6
1.4　基本的なオブジェクト 6
1.5　View の作成 8
1.5.1　XRT/table を使用した View 10
1.5.2　XRT/graph を使用した View 11
1.5.3　XRT/3d を使用した View 12

2　MVC オブジェクト 13

2.1　はじめに 13
2.2　属性 13
2.3　MVC オブジェクトの作成 14
2.4　MVC の手続き 14

3　MVC オブジェクト データとラベル 15

3.1　はじめに 15
3.2　データを MVC オブジェクトに取り込む 15
3.3　データの操作 16
3.4　ラベルの操作 16
3.5　データの書き込み 17
3.6　静的(static)データの使用 17

4　View オブジェクト 19

4.1　はじめに 19

4.2 View クラスの登録 19
4.3 共通の属性 20
4.3.1 Display Policy 20
4.3.2 Range 20
4.3.3 Editability 21
4.4 XRT/table の View 21
4.5 XRT/graph の View 21
4.6 XRT/3d の View 22

5 高度なプログラミング 23

5.1 ソース・コード 23
5.2 任意の View 23
5.2.1 View のイニシャライズ 24
5.2.2 View の破棄 24
5.2.3 編集可・不可の変更 24
5.2.4 View のアップデート 24

Part II リファレンス

A オブジェクトの手続き 29

A.1 MVC オブジェクトの手続き 29
A.2 MVC データの手続き 30
A.3 MVC ラベルの手続き 31
A.4 View オブジェクトの手続き 32
A.4.1 Graph View オブジェクトの手続き 32
A.4.2 3d View オブジェクトの手続き 32
A.4.3 Table View オブジェクトのリソース 33
A.5 ユーティリティ関数 34

B 手続きと方法 35

索引 49

Preface

MVC について ■ 前提知識

マニュアルの概要

MVC について

Model View/Control(MVC)は、XRT/graph、XRT/3d、XRT/table を使用して一つのデータを異なる View で表示するアドオン・ツールです。MVC Kit によって、一つの View で加えられた変更が、元のデータとその他の View に直ちにアップデートされるようになります。

このマニュアルでは、MVC Kit のみを解説します。MVC Kit は、XRT/graph、XRT/3d、XRT/table と一緒に使用されるように設計されていますので、MVC を使用する時は、これらの製品のマニュアルも参考にしてください。

前提知識

このマニュアルは、ユーザがC言語やMotifツールキット、XRT製品(XRT/graph、XRT/3d、XRT/table)に精通していることを前提としています。MotifやXRTアプリケーションをプログラミングするには、C言語の“`char *`配列”、“構造体のポインタ”等を理解している必要があります。

マニュアルの概要

Part Iは、MVC Kitの使用法について解説します。

第1章 “簡単なプログラムの開発”は、MVC Kitを使用した基本的なプログラミングを紹介します。

第2章 “MVCオブジェクト”は、MVCオブジェクトの属性や、方法を解説します。

第3章 “MVCオブジェクトのデータとラベル”は、MVCデータ/ラベルとその方法を解説します。

第4章 “Viewオブジェクト”は、View Objectとそのサブ・タイプの属性や方法を解説します。

第5章 “高度なプログラミング”では、一歩進んだMVC Kitの使用法を紹介します。

Part IIの内容は、技術的なリファレンスが付録として提供されています。

付録A “オブジェクトの手続き”は、オブジェクトと機能にグループ分けされたMVC Kitの手続きの一覧です。

付録B “手続きと方法”は、MVC Kitの全ての手続き、方法、簡易ルーチンのアルファベット順のリストです。

Part I

Using the MVC Kit

1

簡単なプログラムの開発

はじめに ■ 環境変数

プログラムのコンパイルと許可

基本的なオブジェクト ■ View の作成

1.1 はじめに

ここでは、サンプルのプログラムを実際にコンパイル、実行しながらMVC Kitの基本的なプログラミングを紹介していきます。ここで使用するプログラム、*sample1.c* は、MVC オブジェクトにデータをロードし、そのデータをターミナルに表示します。*sample2.c* では、そのデータのViewをXRT/graph、XRT/3d、XRT/tableウィジェットを使用して作成します。

1.2 環境変数

開始する前にUNIXの環境をご確認ください。*XRTHOME*という環境変数が、XRT/graph、XRT/3dまたはXRT/tableのファイルのディレクトリ名として定義されている必要があります。

次のように入力して、*XRTHOME*の現在の値を確認できます。

```
echo $XRTHOME
```

もし*XRTHOME*環境変数が正しく設定されていない場合は、リリース&インストレーション･ガイドを参照するか、システム管理者の方にお問い合わせください。

またMotifのアプリケーションやXRT/graph、XRT/3d、XRT/tableのアプリケーションがコンパイルできることもご確認ください。ご使用になっているMotifのマニュアルで、Motifのライブラリとインクルード･ファイルの位置を調べることができます。MVC Kitの使用を開始される前に、簡単なMotifやXRTのアプリケーションがコンパイルでき稼動することもご確認ください。

1.3 プログラムのコンパイルと許可

*sample1.c*はXRTを必要としないので、許可を与えずにコンパイルするだけで実行できます。但し、*sample2.c*は、下記のようにアプリケーションに許可を与える必要があります。

X、MotifやXRTライブラリとインクルード・ファイルが標準の場所(*/usr/lib*、*/usr/include/X11*、*/usr/include/Xm*)にインストールされていれば、*sample2.c*を次のようにしてコンパイル[1]し、許可を与えることができます。

```
cc sample2.c -I$XRTHOME/include \
  -L$XRTHOME/lib -o sample2 \
  -lxrtmvc -lxrtm -lxrt3d -lxrttable -lXm -lXt -lX11 -lm
$XRTHOME/bin/xrt_auth sample2
XRTHOME:[LIB]XRTMVC.OLB/LIBRARY
```

もし、ライブラリとインクルード・ファイルが標準の位置になければ、`-I`と`-L`のコンパイラ・フラグが必要です。

xrt_auth

XRT を使用したプログラムは、全て *xrt_auth* を使用して許可を与える必要があります。詳細は、それぞれの製品のマニュアルを参照してください。

C++と ANSI C

MVC Kit のインクルード・ファイルは、C++、ANSI C、”Kernighan and Ritchie”C と互換性があります。

1.4 基本的なオブジェクト

図 1 は、*sample1.c* です。*sample1.c* は、ファイルからデータを MVC オブジェクトにロードし、そのデータをターミナルにコピーする最小限の C プログラムです。数値がカンマで区切られた数行のデータ・ファイル、*sample.dat* が作成されます。

[1] コンパイルのオプションは機種に依存します。ご使用のマシンでのコンパイルの例が *$XRTHOME/src/common_demos* の *Makefile* にあります。

```
#include <stdio.h>
#include <Xm/XrtMvc.h>

int
main(int argc, char *argv[])

{
   XrtMvc mvc;
   FILE *fp;
   int rows, columns, i, j;

   /* Create the MVC object. */
   mvc = XrtMvcCreate();

   /* Open the data file. */
   if (!(fp = fopen("sample.dat", "r"))) {
       perror("sample.dat");
       exit(0);
   }

   /* Read the data into the MVC object. */
   XrtMvcDataReadAscii(mvc, fp, ',');
   Fclose(fp);

   /* Dump the MVC data structure to the terminal. */
   columns = XrtMvcDataGetColumnCount(mvc);
   rows = XrtMvcDataGetRowCount(mvc);

   for (i = 0; i < rows; i++) {
       for (j = 0; j < columns; j++) {
           printf("%7.2f ", XrtMvcDataGetValue(mvc, i, j));
       }
   printf("\n");
}
```

図 1　sample1.c

図 2 は、*sample1.c* と *sample2.c* のデータ例です。

```
1.0     2.0     3.0     4.0     5.0
2.0     4.0     6.0     8.0     10.0
3.0     6.0     10.0    16.0    24.0
4.0     8.0     16.0    30.0    40.0
5.0     10.0    24.0    40.0    60.0
```

図 2　sample1.c と sample2.c のデータ

このプログラムがコンパイル、実行されると、図 3 が表示されます。

```
% sample1
1.00    2.00    3.00    4.00    5.00
2.00    4.00    6.00    8.00    10.00
3.00    6.00    10.00   16.00   24.00
4.00    8.00    16.00   30.00   40.00
5.00    10.00   24.00   40.00   60.00
%
```

図 3　sample1.c の出力結果

1.5　View の作成

sample1.c では、XRT ウィジェットが使用されていません。次のプログラム、*sample2.c* では、XRT を使用して View を作成します。

図 4 のコードには、実際に View を作成、登録し、ウィジェットと関連付ける部分は含まれていません。使用している XRT 製品によって View を作成するコードが若干異なりますで、XRT/table、XRT/graph、XRT/3d それぞれに対する例は、図 4 の後に掲載してあります。

図 4 は、各 XRT 製品に共通な *sample2.c* の部分です。

```
#include <stdio.h>
#include <Xm/XrtMvc.h>

int
main(int argc, char *argv[])

{
    Widget      toplevel, view_widget;
    XrtMvcView  view_object;
    XrtMvc      mvc;
    XmString    str;
    FILE        *fp;

    /* Create the MVC object. */
    mvc = XrtMvcCreate();

    /* Open the data file. */
    if (!(fp = fopen("sample.dat", "r"))) {
        perror("sample.dat");
        exit(0);
    }

    /* Read the data into the MVC object. */
    XrtMvcDataReadAscii(mvc, fp, ',');
    fclose(fp);

    /* Initialise Intrinsics. */
    toplevel = XtInitialize(argv[0], "demo", NULL, 0, &argc, argv);

        /*
        ...Product dependent section (see figures below)...
        */

    XtRealizeWidget(toplevel);
    XtMainLoop();
}
```

図 4　sample2.c

1.5.1 XRT/table を使用した View

図 5 のコードでは、*sample2.c* が MVC オブジェクトのデータをテーブル(表)に表示するようにしています。

```
#include <Xm/XrtTable.h>

...

    /* Create a view widget to display the MVC data. */
    view_widget = XtVaCreateManagedWidget("mvc_view",
            xtXrtTableWidgetClass, toplevel,
            NULL);

    /* Register the table view class. */
    XrtMvcViewXrtTableRegister();

    /* Associate a view of the data with the table widget and display it. */
    view_object = XrtMvcViewCreate(mvc, XrtMvcXrtTableViewClass,
            view_widget, NULL);
```

図 5　XRT/table を使用する sample2.c

このプログラムがコンパイル、実行されると、図 6 が表示されます。

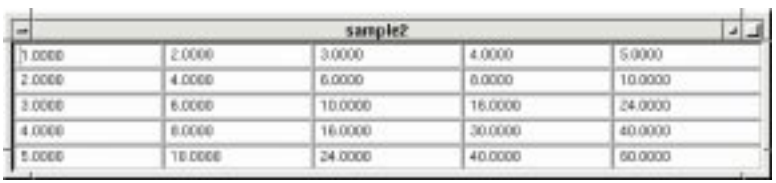


図 6　XRT/table を使用した sample2.c で表示されるウィンドウ

1.5.2 XRT/graph を使用した View

図 7 のコードでは、*sample2.c* が MVC オブジェクトのデータを 2D グラフに表示するようにしています。

```
#include <Xm/XrtGraph.h>

...

    /* Create a view widget to display the MVC data. */
    view_widget = XtVaCreateManagedWidget("mvc_view",
        xtXrtGraphWidgetClass,toplevel,
        XtNxrtType, XRT_TYPE_PLOT,
        NULL);

    /* Register the graph view class. */
    XrtMvcViewXrtGraphRegister();

    /* Associate a view with the graph widget and display it. */
    view_object = XrtMvcViewCreate(mvc, XrtMvcXrtGraphViewClass,
        view_widget, NULL);
```

図 7　XRT/graph を使用する sample2.c

このプログラムがコンパイル、実行されると、図 8 が表示されます。

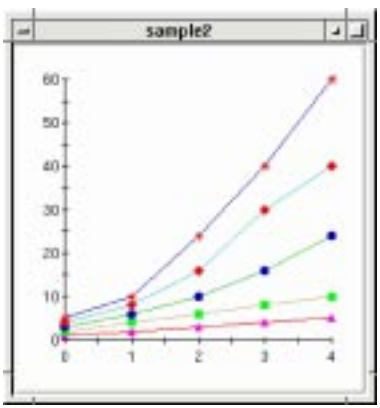


図 8　XRT/graph を使用した sample2.c で表示されるウィンドウ

1.5.3 XRT/3d を使用した View

図 9 のコードでは、*sample2.c* が MVC オブジェクトのデータを 3D サーフェスに表示するようにしています。

```
#include <Xm/Xrt3d.h>

...

    /* Create a view widget to display the MVC data. */
    view_widget = XtVaCreateManagedWidget("mvc_view",
        xtXrt3dWidgetClass,  toplevel,
        XtNxrt3dDrawShaded,  True,
        XtNxrt3dDrawZones,   True,
        NULL);

    /* Register the 3d view class. */
    XrtMvcViewXrt3dRegister();

    /* Associate a view with the 3d widget and display it. */
    view_object = XrtMvcViewCreate(mvc, XrtMvcXrt3dViewClass,
        view_widget, NULL);
```

図 9　XRT/3d を使用する sample2.c

このプログラムがコンパイル、実行されると、図 10 が表示されます。

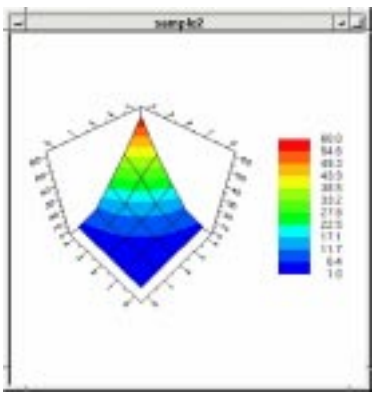


図 10　XRT/3d を使用した sample2.c で表示されるウィンドウ

2

MVC オブジェクト

はじめに ■ 属性

MVC オブジェクトの作成 ■ MVC の手続き

2.1 はじめに

この章では、MVC オブジェクトとその属性や関連する方法等について解説します。MVC オブジェクトは、View オブジェクトで参照されるデータ・スペースや、データを操作する方法、属性等を持っています。尚、データ・スペースと方法は、次の章で解説されています。

Appendix B には、MVC Kit の全ての方法の一覧がありますので参照してください。

2.2 属性

MVC オブジェクトは、幾つかの属性を持ち、これらは(各属性の下に説明されている)方法を使用して取り出したり変更したりすることができます。

`ColumnCount` `integer` `0`

MVC オブジェクトのデータの列数。この属性は read-only で、*XrtMvcDataGetColumnCount()*でアクセスすることができます。

`Update` `Boolean` `True`

アップデートがViewに送られるかどうかを指定します。*XrtMvcGetUpdate()*によってアクセスでき、*XrtMvcSetUpdate()*でセットできます。

`UserData` `XtPointer` `NULL`

MVCオブジェクトと関連付けられるユーザ・データ。*XrtMvcGetUserData()*によってアクセスでき、*XrtMvcSetUserData()*でセットできます。

```
RowCount            integer                 0
```

MVC オブジェクトのデータの行数。この属性は read-only で、*XrtMvcDataGetRowCount()* によってアクセスできます。

Update の属性

`Update` の属性は、MVC データの変更を View に渡すか、またはバッファに入れるかをコントロールします。`True` の場合、変更される度にその変更が View に渡されます。`False` の場合は、全ての変更は、次に `True` になるまでキュー(queue)に入れられます。この方法は、MVC オブジェクトに多くの変更が加えられる場合、View がその度再描画されないようにする時に便利です。`Update` は、*XrtMvcSetUpdate()*を使用してセットできます。

2.3 MVC オブジェクトの作成

*XrtMvcCreate()*は、空の MVC オブジェクト構造体を作成します。*XrtMvcDataReadAscii()* を使用してファイルからデータを読み込むか、または *XrtMvcDataAppendColumn()*、*XrtMvcDataAppendRow()*、*XrtMvcDataSetValue()*を使用して配列を作成し、その構造体にデータを加えてください。View が作成された後にデータを加える場合は、`Update` をオフにしておくと、View が途中の段階で再描画されません。

2.4 MVC の手続き

MVC Kit には、幾つかの手続きが提供されています。これらの手続きには、MVC オブジェクトの変更された範囲が View に表示されている範囲に重なるかを調べる *XrtMvcIntersectRanges()*や、指定されたウィジェットに表示される View を調べる *XrtMvcWidgetToView()*があります。尚、Appendix B には、これらの手続きとその解説の一覧があります。

3

MVC オブジェクト データとラベル

はじめに ■ データを MVC オブジェクトに取り込む

データの操作 ■ ラベルの操作

データの書き込み ■ 静的(static)データの使用

3.1 はじめに

MVC オブジェクトが View を作成するために記憶する情報には、データとラベルの二つのクラスがあります。データは、`double` 型の数値で、これらは行と列の 2 次元配列の構造体に格納されます。ラベルは、リスト構造体に格納される `String` 型です。

MVC データの未定義のエレメントは、`XRTMVC_VALUE_UNDEFINED` と表され、セットされていないエレメントを表示する View に渡すことができます。

3.2 データを MVC オブジェクトに取り込む

MVC オブジェクトにデータを取り込むには、ファイルから読み込む方法が一番簡単です。データが Unix ファイルにあり、リアルタイムに変更またはアップデートされる必要がなければ、このデータを *XrtMvcDataReadAscii()*が理解できるフォーマットにします。次に、*XrtMvcDataReadAscii()*を呼び出してファイルから MVC オブジェクトにこのデータをロードしてください。(*XrtMvcDataReadAscii()*は Appendix B に解説があります。)

その他には、*XrtMvcDataAppendRow()*、*XrtMvcDataAppendColumn()*を使用して MVC オブジェクトに追加される行/列を作成するか、または *XrtMvcDataSetValue()*によってエレメントをセットし、直接 MVC オブジェクトをデータでセットする方法があります。データは、(*fgets(3)*または *fscanf(3)*等を使用して)データ・ファイルから読み込んだり、内部的に生成させたり、他のアプリケーションから提供させることができます。

ラベルは、*XrtMvcLabelAppendRow()*と *XrtMvcLabelInsertRow()*を使用して追加してください。それ以外の方法は使用できません。

3.3 データの操作

MVC Kit には、データ・オブジェクトのデータをセットしたり取り出したりする手続きや方法の他、データを再編成したり削除したりする方法等が提供されています。下の表は、これらの一覧ですが、詳しい解説は Appendix B を参照してください。また、これらの手続きのソース・コードは、*$XRTHOME/src/mvc* にあります。

データの追加と削除	
XrtMvcDataAppendRow()	**データの行を** MVC **オブジェクトに追加**(append)**する**
XrtMvcDataAppendColumn()	**データの列を** MVC **オブジェクトに追加**(append)**する**
XrtMvcDataInsertRow()	**データの行を** MVC **オブジェクトに挿入する**
XrtMvcDataInsertColumn()	**データの列を** MVC **オブジェクトに挿入する**
XrtMvcDataDeleteRows()	**データの行を** MVC **オブジェクトから削除する**
XrtMvcDataDeleteColumns()	**データの列を** MVC **オブジェクトから削除する**
XrtMvcDataDeleteAll()	MVC **オブジェクトから全てのデータを削除する**
データ値の取得と変更	
XrtMvcDataGetValue()	**指定された**(x, y)**にある数値を取り出す**
XrtMvcDataSetValue()	**指定された**(x, y)**にある数値をセットする**
XrtMvcDataMoveRows()	**データ・オブジェクト内で行の位置を変更する**
XrtMvcDataMoveColumns()	**データ・オブジェクト内で列の位置を変更する**

3.4 ラベルの操作

ラベルは、データの場合と同じ手続きを使用してアクセスすることができますが、ファイルの読み込み/書き込みや個別に取得/設定する手続きは使用できません。データを操作してもラベルは影響されませんので、データとラベルを一致させるためには、データとラベルの両方の追加、挿入、移動、削除の手続きを呼び出す必要があります。

これらの手続きの詳しい解説は、Appendix B を参照してください。また、これらの手続きのソース・コードは、*$XRTHOME/src/mvc* にあります。

XrtMvcLabelAppendRow()	**行ラベルを** MVC **オブジェクトの行ラベル・リストにに追加**(append)**する**
XrtMvcLabelAppendColumn()	**列ラベルを** MVC **オブジェクトの列ラベル・リストに追加**(append)**する**
XrtMvcLabelInsertRow()	**行ラベルを** MVC **オブジェクトの行ラベル・リストに挿入する**
XrtMvcLabelInsertColumn()	**列ラベルを** MVC **オブジェクトの列ラベル・リストに挿入する**
XrtMvcLabelDeleteRows()	**行ラベルを** MVC **オブジェクトから削除する**

XrtMvcLabelDeleteColumns()	**列ラベルを** MVC **オブジェクトから削除する**
XrtMvcLabelDeleteAll()	MVC **オブジェクトから全てのラベルを削除する**
XrtMvcLabelMoveRows()	**行ラベル･リスト内で行ラベルの位置を変更する**
XrtMvcLabelMoveColumns()	**列ラベル･リスト内で列ラベルの位置を変更する**

3.5 データの書き込み

MVC データは、*XrtMvcDataWriteAscii()*を使用してテキスト･ファイルに書き込むことができます。*XrtMvcDataReadAscii()*と同様に、この手続きはデータのみを扱いラベルには使用できません。この手続きは、数値をプリントする際に使用する format や行にある数値を区切る delimiter の引数を取ります。

3.6 静的(static)データの使用

表示されるデータがコンパイル時に明らかな場合は、静的変数として定義し、MVC オブジェクトに追加(append)することができます。

$XRTHOME/src/common_demos/mvc/main.c にあるデモ･プログラムでは、静的データを MVC オブジェクトにロードしていますので参照してください。

4

View オブジェクト

はじめに ■ View クラスの登録

共通の属性 ■ XRT/table の View

XRT/graph の View ■ XRT/3d の View

4.1 はじめに

この章では、View オブジェクトの型や属性、それに関連する方法を解説しています。View オブジェクトは、MVC オブジェクトに格納されているデータ(またはその範囲)を持ちます。

View オブジェクトの型は、インスタンスが生成(instantiate)される前に Register の方法を用いて必ず登録してください。

Appendix B には、MVC Kit の方法の一覧がありますので参照してください。

4.2 View クラスの登録

View が作成される前に、必ず View クラスを登録してください。これにより、幾つかの定数が定義され、View を維持する基本的なコールバックを持ったクラス構造体が作成されます。

View クラスを登録するには、*XrtMvcViewXrtTableRegister()* 、*XrtMvcViewXrtGraphRegister()*、*XrtMvcViewXrt3dRegister()*の内のどれかを呼び出してください。*XrtMvcViewRegisterAllClasses()*を使用すると、三つの定義済みの View を一度に登録することができます。

*XrtMvcViewRegisterClass()*は、任意のクラスを定義する時に使用します。View を作成、維持、破棄する全てのコールバックを指定してください。

4.3 共通の属性

全ての View オブジェクトは、次の共通の属性を持ちます。

DisplayPolicy XrtMvcDisplayPolicy XRTMVC_DISPLAY_ALL

表示するデータの範囲。なし(none)、全てまたは指定された範囲です。

Range XrtMvcRange(0,0->XRTMVC_LAST_ROW,XRTMVC_LAST_COLUMN)

範囲が指定された View に表示されるデータの範囲。

Editability Boolean True

View にあるデータが変更できるかどうかを調べます。

UserData XtPointer NULL

XmNuserData と同様に、任意に使用する場合のストーレッジです。

各属性に対して、属性の現在値を取得する *XrtMvcViewGet{attribute name}()*と属性の値を変更する *XrtMvcViewSet{attribute name}()*があります。

4.3.1 Display Policy

DisplayPolicy は、表示する MVC データの範囲を指定します。値は、次の通りです。

XRTMVC_DISPLAY_ALL	全てのデータを表示する
XRTMVC_DISPLAY_RANGE	指定された範囲のデータを表示する
XRTMVC_DISPLAY_NOTHING	データを表示しない

XRTMVC_DISPLAY_RANGE にセットされた場合は、range の属性で表示するデータの範囲を指定してください。

4.3.2 Range

MVC データの範囲を定義し、この範囲にあるデータが View に表示されます。Range を使用する場合は、DisplayPolicy が XRTMVC_DISPLAY_RANGE にセットされている必要があります。データの最後がわからなかったり、変数であっても、XRTMVC_LAST_ROW と XRTMVC_LAST_COLUMN という定数を使用して、配列の最後を指定することができます。

4.3.3 Editability

View のデータに加えられた変更を MVC データに反映させるかどうかをコントロールします。`True` の場合、編集の関数が、(View の変更に)対応する変更を MVC に追加します。`False` の場合は、全ての変更は一時的なものとして処理され、View が次にアップデートまたは再描画された時に上書きされます。

但し、View を編集可能に設定する場合、MVC データ・オブジェクトがバッファリングされていると、そのデータと View のデータが異なってしまいますので注意してください。

4.4 XRT/table の View

XRT/table の View は、再描画時に MVC オブジェクトから直接データを参照します。現在のバージョンでは、(編集可能な)テーブルから MVC データに対する変更は、一つのセルの編集のみがサポートされています。XRT/table のドラッグ&ドロップ機能は、サポートされていません。

Table の View には、次の属性があります。

`DisplayFormat` `String` `"%.4lf"`

表示される時にテーブルのデータに適用されるフォーマット。

`ScanFormat` `String` `"%lf"`

テーブルの入力を調べる時に使用されるフォーマット。

`UndefinedString` `String` `"undefined"`

値が定義されていないセルに表示されるストリング。

`DisplayFormat` は、*printf(3)*スタイルのフォーマット・ストリングを取り、double 型の引数を受け入れる変換文字(conversioin character)が含まれている必要があります。`ScanFormat` は、*scanf(3)*スタイルのフォーマット・ストリングを取り、これには、”`le`”、”`lf`”、”`lg`”の内一つの変換が必要です。

方法の名前の接頭辞は、*XrtMvcViewXrtTableGet{name}()* と *XrtMvcViewXrtTableSet{name}()*です。

4.5 XRT/graph の View

XRT/graph の View には、表示されている MVC データの範囲のコピーが渡されます。従って、DataHandle が使用されている場合、MVC データの変更を反映させずに、データを変更することができます。これらの変更は、View が MVC からアップデートされる時に、上書きされます。

Graph の View は次の属性を持っています。

`XValueRow` `integer` `XRTMVC_UNSPECIFIED_ROW`

列データで、X 軸の位置を定義する行データ。

X 値の行がセットされていない場合は、XRTMVC_UNSPECIFIED_ROW が使用され、グラフの View は、列番号を用いて X 軸のポイントを決定します。この属性がセットされている場合は、グラフのポイントを横方向にドラッグしてその値を変更することができます。

方法には、*XrtMvcViewXrtGraphGetXValuesRow()*と *XrtMvcViewXrtGraphSetXValuesRow()* があります。

4.6 XRT/3d の View

XRT/3d の View にも MVC データのコピーが渡され、データの変更は、XRT/graph の View の場合と同じように処理されます。

3D の View は、次の属性を持っています。

XValuesRow integer XRTMVC_UNSPECIFIED_ROW

列データで、X 軸の位置を定義する行データ。

YValuesColumn integer XRTMVC_UNSPECIFIED_COLUMN

行データで、Y 軸の位置を定義する列データ。

Y 値の列に対する X 値の行がセットされていない場合は、XRTMVC_UNSPECIFIED_ROW が使用されます。(Y 値の列がセットされていない場合は、XRTMVC_UNSPECIFIED_COLUMN が使用されます。)この場合、3D の View は、列番号を用いて X 軸のポイントを(または、行番号を用いて Y 座標を)決定します。

X 値の行と Y 値の列は規則的なグリッドを構成するようにしてください。

方法の名前の接頭辞は、*XrtMvcViewXrt3dGet{name}()*と *XrtMvcViewXrt3dSet{name}()*です。

5

高度なプログラミング

ソース･コード

任意の View

5.1 ソース･コード

MVC Kit には、全ての手続きのソース･コードが提供されています。ライブラリのコードは、ヘッダー･ファイルとは異なり、K&R C とは互換性がありませんので、ANSI に準拠したコンパイラを使用してコンパイルしてください。

ソース･コードは、*$XRTHOME/src/mvc* にあります。

5.2 任意の View

*XrtMvcViewRegisterClass()*を使用して、View をイニシャライズ、破棄、アップデート、編集の可･不可を変更する関数を定義し、任意の View クラスを作成することができます。ヘッダー･ファイル、*XrtMvc.h* に定義されている関数のテンプレートは次の通りです。

```
typedef void (*XrtMvcViewInitializeProc)     View をイニシャライズ
  (XrtMvc mvc, XrtMvcViewStruct *view);

typedef void (*XrtMvcViewDestroyProc)        View を破棄
  (XrtMvc mvc, XrtMvcViewStruct *view);

typedef void (*XrtMvcViewChangeEditProc)     View の編集可･不可をセット
  (XrtMvc mvc, XrtMvcViewStruct *view);

typedef void (*XrtMvcViewUpdateProc)         View をアップデート
  (XrtMvc mvc, XrtMvcViewStruct *view,
  XrtMvcChangeStruct *changes, int num_changed);
```

5.2.1 View のイニシャライズ

View のイニシャライズの関数は、*XrtMvcViewCreate()*によって呼び出されます。この関数は、MVC オブジェクトと、View のポインタを取ります。

この関数は、View 固有の変数を割り当て、セットします。`XrtMvcViewStruct` の `view_data` という属性は、View 固有のデータを保持します。(View の構造体、`XrtMvcViewStruct` は、*XrtMvc.h* に定義されています。)

5.2.2 View の破棄

View の破棄の関数は、*XrtMvcViewDestroy()*または、View を破棄するウィジェットのコールバックによって呼び出されます。イニシャライズの関数と同様、この関数も、MVC オブジェクトと、View のポインタを取ります。

破棄の関数は、イニシャライズの関数で割り当てられていたメモリを解放し、(ファイルを閉じる等の)タスクを実行するようにしてください。

5.2.3 編集可･不可の変更

編集可･不可を変更する関数にも、MVC オブジェクトと、View のポインタが渡されます。この関数は、*XrtMvcViewSetEditability()*によって呼び出され、View の編集可･不可を変更します。

5.2.4 View のアップデート

View のアップデートの関数には、MVC オブジェクトと、View のポインタの他に、MVC データの変更のリストのポインタと変更された数が渡されます。各変更は、次の構造体によって定義されます。

```
typedef struct {
  XrtMvcChangeAction action;
  XrtMvcRange range;
} XrtMvcChangeStruct;

typedef struct {
  int start_row;
  int start_column;
  int end_row;
  int end_column;
} XrtMvcRange;
```

action は、次の値の内のどれかです。

```
XRTMVC_APPENDED_COLUMNS
XRTMVC_APPENDED_COLUMN_LABELS
XRTMVC_APPENDED_ROWS
XRTMVC_APPENDED_ROW_LABELS
XRTMVC_CHANGED_ALL
XRTMVC_DELETED_COLUMNS
XRTMVC_DELETED_COLUMN_LABELS
XRTMVC_DELETED_ROWS
XRTMVC_DELETED_ROW_LABELS
XRTMVC_INSERTED_COLUMNS
XRTMVC_INSERTED_COLUMN_LABELS
XRTMVC_INSERTED_ROWS
XRTMVC_INSERTED_ROW_LABELS
XRTMVC_LABELS_CHANGED
XRTMVC_MOVED_COLUMNS
XRTMVC_MOVED_COLUMN_LABELS
XRTMVC_MOVED_ROWS
XRTMVC_MOVED_ROW_LABELS
XRTMVC_VALUES_CHANGED
```

MVC オブジェクトは、変更の *action* に対して幾つかの最適化を行います。(例: XRTMVC_CHNGED_ALL がバッファリングされている時は、その前にバッファリングされていたアクションを削除する。)

アップデートの関数は、*XrtMvcIntersectRanges()*を使用して、変更の中から、現在 View に表示されている範囲に対する変更だけを捜すことができます。

Part II

Reference Appendices

A

オブジェクトの手続き

MVC オブジェクトの手続き ■ MVC データの手続き
MVC ラベルの手続き ■ View オブジェクトの手続き
ユーティリティ関数

MVC Kit で提供されている手続きを分類してまとめたリストです。各手続きの詳細は、Appendix B に解説されています。

A.1 MVC オブジェクトの手続き

一般的な関数	
XrtMvcCreate()	MVC **オブジェクトを作成する**
取得と変更の属性	
XrtMvcGetViewCount()	MVC **オブジェクトが持っている** View **の数を取り出す**
XrtMvcGetUpdate()	**現在のアップデートの状態を返す**
XrtMvcSetUpdate()	**アップデートの状態をセットする**
XrtMvcGetUserData()	MVC **オブジェクトと関連しているユーザ･データを取り出す**
XrtMvcSetUserData()	MVC **オブジェクトのユーザ･データをセットする**

A.2 MVC データの手続き

データの追加	
XrtMvcDataAppendRow()	データの行を MVC オブジェクトに追加(append)する
XrtMvcDataAppendColumn()	データの列を MVC オブジェクトに追加(append)する
XrtMvcDataInsertRow()	データの行を MVC オブジェクトに挿入する
XrtMvcDataInsertColumn()	データの列を MVC オブジェクトに挿入する
XrtMvcDataReadAscii()	MVC オブジェクトをファイルから読み込んだデータでセットする
データの削除	
XrtMvcDataDeleteRows()	データの行を MVC オブジェクトから削除する
XrtMvcDataDeleteColumns()	データの列を MVC オブジェクトから削除する
XrtMvcDataDeleteAll()	MVC オブジェクトから全てのデータを削除する
データの移動	
XrtMvcDataMoveRows()	データ･オブジェクト内で行の位置を変更する
XrtMvcDataMoveColumns()	データ･オブジェクト内で列の位置を変更する
データ値の取得と変更	
XrtMvcDataGetColumnCount()	MVC オブジェクトのデータの列数を返す
XrtMvcDataGetRowCount()	MVC オブジェクトのデータの行数を返す
XrtMvcDataGetValue()	指定された(x, y)座標にある値を取り出す
XrtMvcDataSetValue()	指定された(x, y)座標にあるデータ値をセットする
XrtMvcDataWriteAscii()	MVC オブジェクトのデータをファイルに書く

A.3 MVC ラベルの手続き

ラベルの追加	
XrtMvcLabelAppendRow()	**行ラベルを** MVC **オブジェクトの行ラベル･リストにに追加**(append)**する**
XrtMvcLabelAppendColumn()	**列ラベルを** MVC **オブジェクトの列ラベル･リストに追加**(append)**する**
XrtMvcLabelInsertRow()	**行ラベルを** MVC **オブジェクトの行ラベル･リストに挿入する**
XrtMvcLabelInsertColumn()	**列ラベルを** MVC **オブジェクトの列ラベル･リストに挿入する**
ラベルの移動	
XrtMvcLabelMoveRows()	**行ラベル･リスト内で行ラベルの位置を変更する**
XrtMvcLabelMoveColumns()	**列ラベル･リスト内で列ラベルの位置を変更する**
ラベルの移動	
XrtMvcLabelDeleteRows()	**行ラベルを** MVC **オブジェクトから削除する**
XrtMvcLabelDeleteColumns()	**列ラベルを** MVC **オブジェクトから削除する**
XrtMvcLabelDeleteAll()	MVC **オブジェクトから全てのラベルを削除する**

A.4 View オブジェクトの手続き

View の作成と破棄	
XrtMvcViewCreate()	View オブジェクトを作成する
XrtMvcViewDestroy()	View オブジェクトを破棄する
ウィジェットとクラスの登録	
XrtMvcViewGetWidget()	View を持つウィジェットを取り出す
XrtMvcRegisterAllClasses()	標準の View クラスを登録する
XrtMvcViewRegisterClass()	任意の View クラスとコールバックを登録する
属性	
XrtMvcViewGetDisplayPolicy()	表示するデータの範囲の決まりを取得する
XrtMvcViewSetDisplayPolicy()	表示するデータの範囲の決まりを設定する
XrtMvcViewGetEditability()	View の編集可･不可を調べる
XrtMvcViewSetEditabiolity()	View の編集可･不可を設定する
XrtMvcViewGetRange()	表示されるデータの範囲を取り出す
XrtMvcViewSetRange()	表示されるデータの範囲をセットする
XrtMvcViewGetUserData()	View と関連する、任意に指定したデータを取り出す
XrtMvcViewSetUserData()	View のユーザ･データをセットする

A.4.1 Graph View オブジェクトの手続き

クラスの登録	
XrtMvcViewXrtGraphRegister()	Graph View クラスとコールバックを登録する
属性	
XrtMvcViewXrtGraphGetXValuesRow()	表示データのX座標を提供する行データを取得する
XrtMvcViewXrtGraphSetXValuesRow()	x 座標の行をセットする

A.4.2 3d View オブジェクトの手続き

クラスの登録	
XrtMvcViewXrt3dRegister()	3d View クラスとコールバックを登録する

属性	
XrtMvcViewXrt3dGetXValuesRow()	表示データのX座標を提供する行データを取得する
XrtMvcViewXrt3dSetXValuesRow()	x 座標の行をセットする
XrtMvcViewXrt3dGetYValuesColumn()	表示データの y 座標を提供する列データを取得する
XrtMvcViewXrt3dSetYValuesColumn()	y 座標の列をセットする

A.4.3 Table View オブジェクトのリソース

クラスの登録	
XrtMvcViewXrtTableRegister()	Table View クラスとコールバックを登録する
属性	
XrtMvcViewXrtTableGetDisplayFormat()	表示されているデータのフォーマットに使用されるストリングを取り出す
XrtMvcViewXrtTableSetDisplayFormat()	表示されるデータ値のフォーマット･ストリングをセットする
XrtMvcViewXrtTableGetScanFormat()	テーブルのセルに入力された値を解釈する時に使用されるフォーマットを取り出す
XrtMvcViewXrtTableSetScanFormat()	テーブルのセルに入力された値を解釈する時に使用されるフォーマット･ストリングをセットする
XrtMvcViewXrtTableGetUndefinedString()	未定義の値が表示される時に使用されるストリングを取り出す
XrtMvcViewXrtTableSetUndefinedString()	データの未定義のエレメントに対して表示されるストリングをセットする

A.5 ユーティリティ関数

ユーティリティ関数	
XrtMvcWidgetToView()	特定のウィジェットと関連している View を取り出す
XrtMvcIntersectRange()	二つの範囲が重なるか調べる

B

手続きと方法

MVC Kitの手続きと方法のアルファベット順のリストです。手続きのソース･コードは、*$XRTHOME/src/mvc*にあります。ソースを変更する時は、他のディレクトリにコピーしてください。(名前を変更しない場合は、MVCライブラリの前にリンクしてください。)

XrtMvcCreate()

MVC オブジェクトを作成し、返します。

```
XrtMvc
XrtMvcCreate(void)
```

XrtMvcDataAppendColumn()

MVC オブジェクトのデータに列を追加します。

```
int
XrtMvcDataAppendColumn(
  XrtMvc      mvc,
  double      *values,
  int         num_values
  )
```

double 型の *values* の配列が、新しい列として MVC データの最終列の後に追加されます。

新しい列の数が返されます。

XrtMvcDataAppendRow()

MVC オブジェクトのデータに行を追加します。

```
int
XrtMvcDataAppendRow(
  XrtMvc      mvc,
  double      *values,
  int         num_values
  )
```

values の配列が、新しい行として MVC データの最後に追加されます。

戻り値は、MVC データ構造体の新しい行の位置です。

XrtMvcDataDeleteAll()

MVC オブジェクト内の全てのデータが削除されます。

```
void
XrtMvcDataDeleteAll(
  XrtMvc      mvc
  )
```

この手続きは、データのみを削除するので、ラベルは別に削除してください。

XrtMvcDataDeleteColumns()

MVC オブジェクトから列データを削除します。

```
void
XrtMvcDataDeleteColumns(
  XrtMvc      mvc,
  int         start_column,
  int         num_columns
  )
```

start_column から開始された *num_columns* 数の列が MVC データ構造体から削除されます。

XrtMvcDataDeleteRows()

MVC オブジェクトから行データを削除します。

```
void
XrtMvcDataDeleteRows(
  XrtMvc      mvc,
  int         start_row,
  int         num_rows
  )
```

start_row から開始された *num_rows* 数の行が MVC データ構造体から削除されます。

XrtMvcDataGetColumnCount()

データの列数を返します。(一番長い行の長さに相当します。)

```
int
XrtMvcDataGetColumnCount(
  XrtMvc      mvc
  )
```

XrtMvcDataGetRowCount()

MVC オブジェクトが持つデータの行数を返します。

```
int
XrtMvcDataGetRowCount(
  XrtMvc      mvc
  )
```

XrtMvcDataGetValue()

MVC データ内の*(row, column)*の位置にある値を返します。

```
double
XrtMvcDataGetValue(
  XrtMvc      mvc,
  int         row,
  int         column
  )
```

XrtMvcDataInsertColumn()

MVC データに列を挿入します。

```
void
XrtMvcDataInsertColumn(
  XrtMvc      mvc,
  int         target_column,
  double      *values,
  int         num_values
  )
```

num_values の長さの *values*(double 型の配列)が、新しい列として MVC データの *target_column* の位置に挿入されます。

XrtMvcDataInsertRow()

MVC データに行を挿入します。

```
void
XrtMvcDataInsertRow(
  XrtMvc      mvc,
  int         target_row,
  double      *values,
  int         num_values
  )
```

num_values の長さの *values*(double 型の配列)が、新しい行として MVC データの *target_row* の位置に挿入されます。

XrtMvcDataMoveColumns()

新しい位置に列を移動します。

```
void
XrtMvcDataMoveColumns(
  XrtMvc      mvc,
  int         target_column,
  int         source_column,
  int         num_columns
  )
```

source_column から開始される *num_columns* 数の列が、*target_column* から開始される位置に移動します。

XrtMvcDataMoveRows()

新しい位置に行を移動します。

```
void
XrtMvcDataMoveRows(
  XrtMvc      mvc,
  int         target_row,
  int         source_row,
  int         num_rows
  )
```

source_row から開始される *num_rows* 数の行が、*traget_row* から開始される位置に移動します。

XrtMvcDataReadAscii()

ASCII ファイルから、MVC オブジェクトのデータを読み込みます。

```
void
XrtMvcDataReadAscii(
  XrtMvc      mvc,
  FILE        *fp,
  char        delimiter
  )
```

開かれているファイル *fp* が End-of-File まで読み込まれます。*delimiter* 文字で区切られるファイルの各行が、データの一行分として MVC データに追加されます。MVC にデータが既に存在する場合は、これらの行は既存のデータの最後に追加されます。

ラベルは、この方法では読み込むことはできません。

XrtMvcDataSetValue()

MVC データの*(row, column)*の位置にあるセルを *value* にセットします。

```
void
XrtMvcDataSetValue(
  XrtMvc      mvc,
  int         row,
  int         column,
  double      value
  )
```

XrtMvcDataWriteAscii()

MVC データ構造体の内容を ASCII ファイルに書きます。

```
Boolean
XrtMvcDataWriteAscii(
  XrtMvc          mvc,
  FILE            *fp,
  char            delimiter,
  String          format,
  XrtMvcRange     *range_ptr
  )
```

データは、*fp* ファイルに(各データの行がファイルの一行分として)書かれます。各値は、*printf* スタイルの *format* に従ってフォーマットされ、それぞれ *delimiter* で区切られます。

XrtMvcGetUpdate()

MVC オブジェクトのアップデートの状態を返します。

```
Boolean
XrtMvcGetUpdate(
  XrtMvc      mvc
  )
```

`True` は、データの変更時に MVC が View にアップデートを指示することを示します。`False` の場合は、アップデートは、次に `True` にセットされるまで queue に入ります。

XrtMvcGetUserData()

MVC オブジェクトのユーザ･データのポインタを返します。

```
XtPointer
XrtMvcGetUserData(
  XrtMvc      mvc
  )
```

XrtMvcGetViewCount()

現在、MVC と関連している View の数を返します。

```
int
XrtMvcGetViewCount(
  XrtMvc      mvc
  )
```

XrtMvcIntersectRanges()

指定された二つの範囲の重なる部分を調べます。

```
Boolean
XrtMvcIntersectRanges(
  XrtMvcRange     *range1,
  XrtMvcRange     *range2,
  XrtMvcRange     *overlap_range
  )
```

range1 と *range2* が重なるか調べられます。重なる場合は `True` が返され、*overlap_range* が重なる部分を持つようになります。

XrtMvcLabelAppendColumn()

列ラベルのリストの最後に、*column_label* を追加します。

```
void
XrtMvcLabelAppendColumn(
  XrtMvc          mvc,
  String          column_label
  )
```

XrtMvcLabelAppendRow()

行ラベルのリストの最後に *row_label* を追加します。

```
void
XrtMvcLabelAppendRow(
  XrtMvc          mvc,
  String          row_label
  )
```

XrtMvcLabelDeleteAll()

MVC オブジェクトから全ての行と列ラベルが削除されます。

```
void
XrtMvcLabelDeleteAll(
  XrtMvc      mvc
  )
```

XrtMvcLabelDeleteColumns()

列ラベルを削除します。

```
void
XrtMvcLabelDeleteColumns(
  XrtMvc      mvc,
  int         start_column,
  int         num_labels
  )
```

start_column から開始される *num_labels* 数の列ラベルが削除されます。

XrtMvcLabelDeleteRows()

行ラベルを削除します。

```
void
XrtMvcLabelDeleteRows(
  XrtMvc      mvc,
  int         start_row,
  int         num_labels
  )
```

start_row から開始される *num_rows* 数の行ラベルが削除されます。

XrtMvcALabelInsertColumn()

target_column の位置に *label* を挿入します。

```
void
XrtMvcLabelInsertColumn(
  XrtMvc      mvc,
  int         target_column,
  String      label
  )
```

XrtMvcLabelInsertRow()

target_row の位置に *label* を挿入します。

```
void
XrtMvcLabelInsertRow(
  XrtMvc      mvc,
  int         target_row,
  String      label
  )
```

XrtMvcLabelMoveColumns()

列ラベルを移動します。

```
void
XrtMvcLabelMoveColumns(
  XrtMvc      mvc,
  int         target_column,
  int         source_column,
  int         num_columns
  )
```

source_columns から開始される *num_columns* 数の列ラベルが、*target_column* 位置に移動します。

XrtMvcLabelMoveRows()

行ラベルを移動します。

```
void
XrtMvcLabelMoveRows(
  XrtMvc      mvc,
  int         target_row,
  int         source_row,
  int         num_rows
  )
```

source_rows から開始される *num_rows* 数の行ラベルが *target_row* 位置に移動します。

XrtMvcSetUpdate()

View のアップデートの決まりをセットします。

```
void
XrtMvcSetUpdate(
  XrtMvc      mvc,
  Boolean     update
  )
```

MVC のアップデートの属性が `False` の場合は、変更は View に送られる代りにバッファに入ります。`True` にセットされれば、変更は直ちに View に送られます。

XrtMvcSetUserData()

MVC オブジェクトのユーザ･データをセットします。

```
void
XrtMvcSetUserData(
  XrtMvc      mvc,
  XtPointer   user_data
  )
```

XrtMvcViewCreate()

新しい View オブジェクトを作成します。

```
XrtMvcView
XrtMvcViewCreate(
  XrtMvc              mvc,
  XrtMvcViewClass     view_class,
  Widget              widget,
  XtPointer           view_user_data
  )
```

view_class クラスの View オブジェクトが作成され、*widget* と関連付けられます。*view_user_data* が、View の初期化の手続きに渡されます。

XrtMvcViewDestroy()

View オブジェクトを破棄します。

```
void
XrtMvcViewDestroy(
  XrtMvcView          view
  )
```

XrtMvcViewGetDisplayPolicy()

View の表示の決まり(Display Policy)を返します。

```
XrtMvcDisplayPolicy
XrtMvcViewGetDisplayPolicy(
  XrtMvcView          view
  )
```

Display Policy の値は、次の通りです。

XRTMVC_DISPLAY_ALL	全てのデータを表示する
XRTMVC_DISPLAY_RANGE	指定された範囲のデータを表示する
XRTMVC_DISPLAY_NOTHING	データを表示しない

XrtMvcViewGedEditability()

View のデータが編集可能であれば、True を返します。

```
Boolean
XrtMvcViewGetEditability(
  XrtMvcView          view
  )
```

XrtMvcViewGetRange()

View が表示する MVC データの範囲を返します。

```
XrtMvcRange
*XrtMvcViewGetRange(
  XrtMvcView          view
  )
```

XrtMvcViewGetUserData()

View のユーザ・データのポインタを返します。

```
XtPointer
XrtMvcViewGetUserData(
  XrtMvcView      view
  )
```

注意: これは、*XrtMvcViewCreate()*手続きでセットされる *view_user_data* と同じではありません。

XrtMvcViewGetWidget()

View と関連しているウィジェットを返します。

```
Widget
XrtMvcViewGetWidget(
  XrtMvcView      view
  )
```

XrtMvcViewRegisterAllClasses()

全ての View オブジェクト・クラスを登録します。

```
void
XrtMvcViewRegisterAllClasses(void)
```

XrtMvcViewRegisterClass()

独自に定義した View オブジェクト・クラスを登録します。

```
XrtMvcViewClass
XrtMvcViewRegisterClass(
  XrtMvcViewInitializeProcinitialize,
  XrtMvcViewDestroyProc       destroy,
  XrtMvcViewUpdateProc        update,
  XrtMvcViewChangeEditProcchange_edit
  )
```

initialize、*destroy*、*update*、*change_edit* は、定義された View をコントロールする時に MVC が使用する関数です。

XrtMvcViewSetDisplayPolicy()

View の表示の決まり(Display Policy)をセットします。

```
void
XrtMvcViewSetDisplayPolicy(
  XrtMvcView          view,
  XrtMvcDisplayPolicy display_policy
  )
```

Display Policy は、View によって表示される MVC データの範囲をコントロールします。値は、次の通りです。

XRTMVC_DISPLAY_ALL	全てのデータを表示する
XRTMVC_DISPLAY_RANGE	指定された範囲のデータを表示する
XRTMVC_DISPLAY_NOTHING	データを表示しない

XrtMvcViewSetEditability()

View の編集可･不可の属性をセットします。

```
void
XrtMvcViewSetEditability(
  XrtMvcView      view,
  Boolean         editable
  )
```

True(View の編集可)の場合は、View の値は変更することができます。

XrtMvcViewSetRange()

View の範囲の属性をセットします。

```
void
XrtMvcViewSetRange(
  XrtMvcView      view,
  XrtMvcRange     *range
  )
```

range は、display policy が XRTMVC_DISPLAY_RANGE にセットされている時に、View に表示される MVC データの範囲を定義します。

XrtMvcViewSetUserData()

View のユーザ･データの属性をセットします。

```
void
XrtMvcViewSetUserData(
  XrtMvcView      view,
  XtPointer       user_data
  )
```

XrtMvcViewXrt3dGetXValuesRow()

XRT/3d の View でグリッドの X 座標に使用される、MVC データの行を返します。

```
int
XrtMvcViewXrt3dGetXValuesRow(
  XrtMvcView      view
  )
```

この属性がセットされていない場合は、XRTMVC_UNSPECIFIED_ROW が使用され、X 座標は、MVC データ配列の列番号になります。

XrtMvcViewXrt3dGetYValuesColumn()

XRT/3d の View でグリッドの Y 座標に使用される、MVC データの列を返します。

```
int
XrtMvcViewXrt3dGetYValuesColumn(
  XrtMvcView      view
  )
```

この属性がセットされていない場合は、XRTMVC_UNSPECIFIED_COLUMN が使用され、Y 座標は、MVC データ配列の行番号になります。

XrtMvcViewXrt3dRegister()

XRT/3d の View オブジェクトを登録します。

```
void
XrtMvcViewXrt3dRegister(void)
```

XrtMvcViewXrt3dSetXValuesRow()

XRT/3d の View でグリッド・ポイントの X 座標を定義する、MVC データの行をセットします。

```
void
XrtMvcViewXrt3dSetXValuesRow(
  XrtMvcView      view,
  int             row
  )
```

この属性を XRTMVC_UNSPECIFIED_ROW とし、X 座標が列番号に基づくようにも設定できます。

XrtMvcViewXrt3dSetYValuesColumn()

XRT/3d の View でグリッド・ポイントの Y 座標を定義する、MVC データの列をセットします。

```
void
XrtMvcViewXrt3dSetYValuesColumn(
  XrtMvcView      view,
  int             column
  )
```

この属性を XRTMVC_UNSPECIFIED_COLUMN とし、Y 座標が行番号に基づくようにも設定できます。

XrtMvcViewXrtGraphGetXValuesRow()

XRT/graph の View でグラフ・ポイントの X 座標を定義する、MVC データの行を取得します。

```
int
XrtMvcViewXrtGraphGetXValuesRow(
  XrtMvcView      view
  )
```

この属性がセットされていない場合は、XRTMVC_UNSPECIFIED_ROW が使用され、X 座標は、MVC データ配列の列番号になります。

XrtMvcViewXrtGraphRegister()

XRT/graph の View オブジェクトを登録します。

```
void
XrtMvcViewXrtGraphRegister(void)
```

XrtMvcViewXrtGraphSetXValuesRow()

XRT/graph の View でグラフの X 座標を定義する、MVC データの行をセットします。

```
void
XrtMvcViewXrtGraphSetXValuesRow(
  XrtMvcView      view,
  int             row
  )
```

この属性を XRTMVC_UNSPECIFIED_ROW とし、X 座標が列番号に基づくようにも設定できます。

XrtMvcViewXrtTableGetDisplayFormat()

テーブルに表示される MVC データのフォーマットに使用されるストリングを返します。

```
String
XrtMvcViewXrtTableGetDisplayFormat(
XrtMvcView        view
)
```

XrtMvcViewXrtTableGetScanFormat()

入力された値を調べる時に、table の View が使用するストリングを返します。

```
String
XrtMvcViewXrtTableGetScanFormat(
  XrtMvcView      view
  )
```

XrtMvcViewXrtTableGetUndefinedString()

テーブルの未定義の値の代りに表示されるストリングを返します。

```
String
XrtMvcViewXrtTableGetUndefinedString(
  XrtMvcView      view
  )
```

XrtMvcViewXrtTableRegister()

XRT/table の View オブジェクトを登録します。

```
void
XrtMvcViewXrtTableRegister(void)
```

XrtMvcViewXrtTableSetDisplayFormat()

テーブルの表示時に、値のフォーマットに使用されるストリングをセットします。

```
void
XrtMvcViewXrtTableSetDisplayFormat(
  XrtMvcView      view,
  String          str
  )
```

str は、table の View に表示される値をフォーマットする、*printf* スタイルのフォーマット･ストリングです。

XrtMvcViewXrtTableSetScanFormat()

ユーザによって入力された値を調べる時に使用されるストリングをセットします。

```
void
XrtMvcViewXrtTableSetScanFormat(
  XrtMvcView      view,
  String          str
  )
```

str は、テーブルのセルに入力された値から数値を取り出す時に使用される、*scanf* スタイルのフォーマット･ストリングです。

XrtMvcViewXrtTableSetUndefinedString()

テーブルで未定義の値の代りに表示されるストリングをセットします。

```
void
XrtMvcViewXrtTableSetUndefinedString(
  XrtMvcView      view,
  String          str
  )
```

XrtMvcWidgetToView()

指定されたウィジェットに表示されている View を調べます。

```
XrtMvcView
XrtMvcWidgetToView(
  Widget          w,
  XrtMvc          *mvc
  )
```

w(ウィジェット)に表示されている View を返し、その View を持つ MVC 構造体を指すように *mvc* をセットします。ウィジェットが View を持っていない時は、戻り値は NULL です。

索引

C

C++, 6

M

MVC オブジェクト
  作成, 14
  属性, 13
  データ型, 15
  データの操作, 16
  データの取り込み, 15
  手続き, 14, 29
  ラベルの型, 15
  ラベルの操作, 16
  ラベルの追加, 15
MVC データ
  手続き, 30

V

View オブジェクト
  Display Policy, 20
  XRT/3d, 22
  XRT/3d の例, 12
  XRT/graph, 21
  XRT/graph の例, 11
  XRT/table, 21
  XRT/table の例, 10
  クラスの登録, 19
  作成, 8
  属性, 20
  手続き, 32
  編集可･不可(editability), 21
  任意, 23

X

XRTHOME, 5
XrtMvcALabelInsertColumn(), 40
XrtMvcCreate(), 14, 35
XrtMvcDataAppendColumn(), 14, 15, 35
XrtMvcDataAppendRow(), 14, 15, 35
XrtMvcDataDeleteAll(), 36
XrtMvcDataDeleteColumns(), 36
XrtMvcDataDeleteRows(), 36
XrtMvcDataGetColumnCount(), 13, 36
XrtMvcDataGetRowCount(), 14, 36
XrtMvcDataGetValue(), 37
XrtMvcDataInsertColumn(), 37
XrtMvcDataInsertRow(), 37
XrtMvcDataMoveColumns(), 37
XrtMvcDataMoveRows(), 38
XrtMvcDataReadAscii(), 14, 15, 38
XrtMvcDataSetValue(), 14, 15, 38
XrtMvcDataWriteAscii(), 38
XrtMvcGetUpdate(), 13, 39
XrtMvcGetUserData(), 13, 39
XrtMvcGetViewCount(), 39
XrtMvcIntersectRanges(), 25, 39
XrtMvcLabelAppendColumn(), 39
XrtMvcLabelAppendRow(), 15, 39
XrtMvcLabelDeleteAll(), 40
XrtMvcLabelDeleteColumns(), 40
XrtMvcLabelDeleteRows(), 40
XrtMvcLabelInsertRow(), 15, 40
XrtMvcLabelMoveColumns(), 41
XrtMvcLabelMoveRows(), 41
XrtMvcSetUpdate(), 13, 14, 41
XrtMvcSetUserData(), 13, 41
XrtMvcViewCreate(), 24, 42
XrtMvcViewDestroy(), 24, 42
XrtMvcViewGedEditability(), 42
XrtMvcViewGetDisplayPolicy(), 42
XrtMvcViewGetRange(), 42
XrtMvcViewGetUserData(), 43
XrtMvcViewGetWidget(), 43
XrtMvcViewRegisterAllClasses(), 19, 43
XrtMvcViewRegisterClass(), 19, 23, 43
XrtMvcViewSetDisplayPolicy(), 43
XrtMvcViewSetEditability(), 24, 44
XrtMvcViewSetRange(), 44
XrtMvcViewSetUserData(), 44
XrtMvcViewXrt3dGetXValuesRow(), 44
XrtMvcViewXrt3dGetYValuesColumn(), 44

XrtMvcViewXrt3dRegister(), 19, 45
XrtMvcViewXrt3dSetXValuesRow(), 45
XrtMvcViewXrt3dSetYValuesColumn(), 45
XrtMvcViewXrtGraphGetXValuesRow(), 45
XrtMvcViewXrtGraphRegister(), 19, 45
XrtMvcViewXrtGraphSetXValuesRow(), 46
XrtMvcViewXrtTableGetDisplayFormat(), 46
XrtMvcViewXrtTableGetScanFormat(), 46
XrtMvcViewXrtTableGetUndefinedString(), 46
XrtMvcViewXrtTableRegister(), 19, 46
XrtMvcViewXrtTableSetDisplayFormat(), 46
XrtMvcViewXrtTableSetScanFormat(), 47
XrtMvcViewXrtTableSetUndefinedString(), 47
XrtMvcWidgetToView(), 47

あ

アプリケーションの許可, 6

か

環境変数, 5

た

データ
  静的, 17
  直接セットする, 15
  ファイルから読む, 15
  ファイルに書く, 17
  未定義, 15