

Xpress アプリケーションディベロッパー

リファレンスマニュアル

リリース 1.0.0

最終更新日 2005 年 5 月 11 日

Chapter 1	1
Mosel integration	1
Chapter 2	3
GUI creation with XAD	3
Chapter 3	5
Lifetime of XAD objects	5
Chapter 4	6
Events	6
Chapter 5	7
XAD objects reference	7
5.1 Window	7
5.2 Text	8
5.3 Simple button	8
5.4 Input	8
5.5 GUI editor	9
5.6 Check button	9
5.7 Radio button	10
5.8 Group of XAD objects	10
5.9 List of items	11
5.10 Progress bar	11
5.11 Tab selectors	12
5.12 List object with multiple columns	12
5.13 Canvas	13
5.14 Browser	13
5.15 Subroutines specific to objects	14
XADcreatewindow	17
XADwindowopen	18
XADwindowclose	19
XADwindowshow	20
XADwindowhide	21
XADwindowkeep	22
XADwindowsettimer	23
XADcreatetext	24
XADtextaddtext	25
XADtextsettext	26

XADtextgettext	27
XADcreatebutton	28
XADcreateinput	29
XADinputsettext	30
XADinputgettext	31
XADcreateeditor	32
XADeditoraddtext	33
XADeditorload	34
XADeditorsave	35
XADeditorsettext	36
XADeditorgettext	37
XADcreatecheck	38
XADchecksetstate	39
XADcheckgetstate	40
XADcreateradio	41
XADradiosetstate	42
XADradiogetstate	43
XADcreategroup	44
XADcreatelist	45
XADlistadd	46
XADlistgetsel	47
XADlistshow	48
XADcreateprogress	49
XADprogressset	50
XADcreatetab	51
XADcreatemultilist	52
XADmultilistshow	53
XADmultilistsetsize	54
XADmultilistsetcolname	55
XADmultilistsettext	56
XADcreatecanvas	57
XADcanvasdrawbox	58
XADcanvasdrawellipse	59
XADcanvaserase	60
XADcanvasrefresh	61
XADcanvasdrawimage	62

XADcanvassaveimage	63
XADcanvasdrawline	64
XADcanvasdrawpoint	65
XADcanvasdrawrectangle	66
XADcanvasdrawtext	67
XADcanvasmap	68
XADcanvasunmap	70
XADcolor	71
XADcreatebrowser	72
XADbrowsergoto	73
5.16 Events specific to objects	74
XAD_EVENT_TIMER	74
XAD_EVENT_WINDOW_CLOSED	75
XAD_EVENT_WINDOW_CLOSING	75
XAD_EVENT_WINDOW_HIDDEN	75
XAD_EVENT_WINDOW_MOVED	75
XAD_EVENT_WINDOW_OPENED	75
XAD_EVENT_WINDOW_RESIZED	75
XAD_EVENT_WINDOW_SHOWN	76
XAD_EVENT_PRESSED	76
XAD_EVENT_CHANGED	76
XAD_EVENT_SELECTION	76
Chapter6	77
Generic routines	77
XADdestroy	78
XADenable	79
XADgetx	80
XADgety	81
XADgetw	82
XADgeth	83
XADrefresh	84
XADsetfocus	85
XADsetpos	86
XADsettext	87
XADsetvisible	88
XADgeteventtext	89

<u>Chapter7</u>	<u>90</u>
<u>Generic events</u>	<u>90</u>
<u>XAD_EVENT_KEYDOWN</u>	<u>90</u>
<u>XAD_EVENT_KEYUP</u>	<u>91</u>
<u>XAD_EVENT_MOUSE_LEFTDOWN</u>	<u>91</u>
<u>XAD_EVENT_MOUSE_LEFTUP</u>	<u>91</u>
<u>XAD_EVENT_MOUSE_MOVED</u>	<u>91</u>
<u>XAD_EVENT_MOUSE_RIGHTDOWN</u>	<u>92</u>
<u>XAD_EVENT_MOUSE_RIGHTUP</u>	<u>92</u>
<u>Chapter8</u>	<u>93</u>
<u>Utility routines</u>	<u>93</u>
<u>XADseteventcallback</u>	<u>94</u>
<u>XADhandleevents</u>	<u>95</u>
<u>XADchoosefile</u>	<u>96</u>
<u>XADpopupmenu</u>	<u>97</u>
<u>Chapter 9</u>	<u>98</u>
<u>Example</u>	<u>98</u>

Chapter 1

Mosel integration

Xpress アプリケーションディベロッパー(XAD)は、Xpress-Mosel モデリングおよびプログラミング言語を拡張したものです。Xpress アプリケーションディベロッパーは Mosel の機能を拡張し、標準ユーザーインターフェースを作成するための一連の関数およびプロシージャを備えています。このため Mosel は、問題の数学表現からユーザーインターフェースの開発まで、完全に最適化されたアプリケーション開発のためのモデリングおよびプログラミング言語として使用することができます。

Xpress アプリケーションディベロッパーは、最適化問題の試験の際、多くの時間を節約することができます。OR 開発者はもはや、ユーザーアプリケーションを構築するために、Mosel と VB、C++ または Java とのインターフェースを必要としないからです。GUI ベースのアプリケーション開発に必要な機能のほとんどが、Mosel 言語の強力かつ使いやすい抽象概念を通して使用できます。

例えば、以下のウィンドウは(IVE の最適化マトリクスダイアログのコピー):

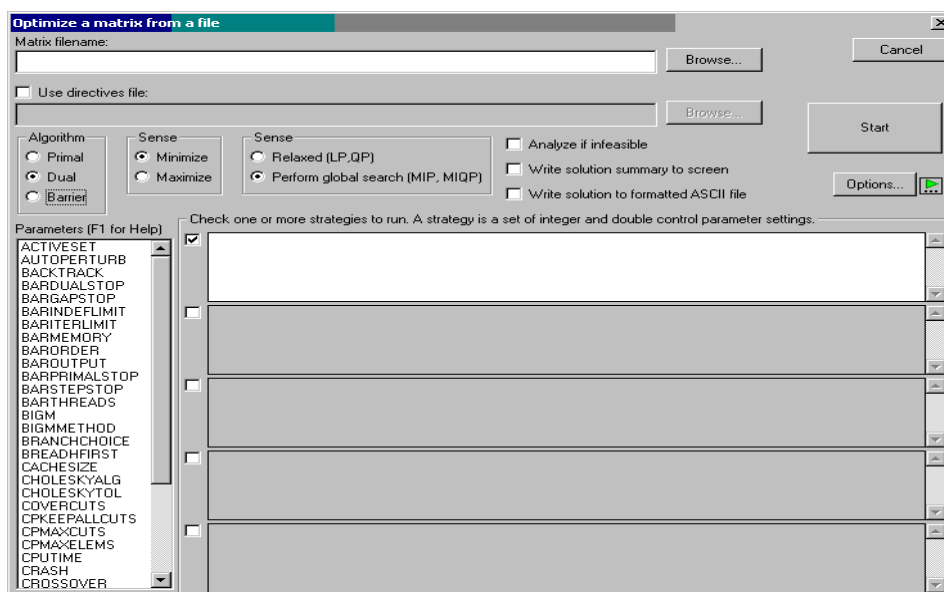


図 1.1: 最適化マトリクスダイアログ

以下の Mosel XAD コードを使用して、Mosel のみで作成したものです:

```

XADcreatewindow      (id_win,                50,50,740,590,"Optimize a matrix from a file")
XADcreatetext       (id_win,id_textmatrix,    3,5,500,20,"Matrix filename:")
XADcreateinput      (id_win,id_inputmatrix,   3,25,500,20,"")
XADcreatebutton     (id_win,id_buttonbrowsematrix, 510,25,75,24,"Browse...")
XADcreatecheck      (id_win,id_checkdirectives, 3,50,120,20,"Use directives file:")
XADcreateinput      (id_win,id_inputdirectives, 3,70,500,20,"")
XADcreatebutton     (id_win,id_buttonbrowstdirectives, 510,68,75,24,"Browse...")
XADcreatebutton     (id_win,id_buttoncancel,    655,5,75,24,"Cancel")
XADcreatebutton     (id_win,id_buttonstart,    620,70,105,51,"Start")
XADcreatebutton     (id_win,id_buttonoptions,   640,140,65,24,"Options...")
XADcreatecanvas     (id_win,id_canvasopticon,   707,145,16,15)
XADcreategroup      (id_win,id_groupalgorithm,  5,98,70,85,"Algorithm")
  XADcreateradio    (id_win,id_radioprimal,    10,115,50,20,"Primal")
  XADcreateradio    (id_win,id_radiodual,     10,135,50,20,"Dual")
  XADcreateradio    (id_win,id_radiobarrier,   10,155,50,20,"Barrier")
XADcreategroup      (id_win,id_groupssense,    90,98,75,65,"Sense")
  XADcreateradio    (id_win,id_radiominimize, 95,115,65,20,"Minimize")
  XADcreateradio    (id_win,id_radiomaximize, 95,135,65,20,"Maximize")
XADcreategroup      (id_win,id_groupoptimization, 180,98,195,65,"Sense")
  XADcreateradio    (id_win,id_radiorelaxed,  185,115,165,20,"Relaxed (LP,QP)")
  XADcreateradio    (id_win,id_radioglobal,   185,135,185,20,"Perform global search (MIP, MIQP)")
XADcreatecheck      (id_win,id_checkanalyze,   385,102,195,20,"Analyze if infeasible")
XADcreatecheck      (id_win,id_checksolsum,   385,127,195,20,"Write solution summary to screen")
XADcreatecheck      (id_win,id_checksolfile,  385,152,195,20,"Write solution to formatted ASCII file")
XADcreatelist       (id_win,id_listparameters, 3,190,125,370,"Parameters (F1 for Help)")
XADcreategroup      (id_win,id_groupstrategies, 130,180,602,384,"Check one or more strategies to run."+
  " A strategy is a set of integer and double control parameter
  forall(i in 0..4) XADcreatecheck(id_win,id_checkstrategy+(i+1), 135,200+73*i,15,15,"")
  forall(i in 0..4) XADcreateeditor(id_win,id_editorstrategy+(i+1), 153,200+73*i,575,70,"")

procedure guievents(id:integer,event:integer)
  if id=id_win and event=XAD_EVENT_WINDOW_OPENED then
    XADcanvasdrawimage(id_canvasopticon,0,0,"runoptions.bmp")
    XADcanvasrefresh(id_canvasopticon)
    XADradiosetstate(id_radiodual,true)
    XADradiosetstate(id_radiominimize,true)
    XADradiosetstate(id_radioglobal,true)
    XADenable(id_inputdirectives,false)
    XADenable(id_buttonbrowstdirectives,false)
    XADchecksetstate(id_checkstrategy+1,true)
    forall(i in 2..5) XADenable(id_editorstrategy+i,false)
    XADlistshow(id_listparameters,{"ACTIVESET","AUTOPERTURB","BACKTRACK","BARDUALSTOP","BARGAPSTOP","BARINDEFI
    "BARPRIMALSTOP","BARSTEPSTOP","BARTHREADS","BIGM","BIGMETHOD","BRANCHCHOICE","BREADHFIRST","CACHESIZE","C
    "CPMAXELEM","CPUTIME","CRASH","CROSSOVER","CUTDEPTH","CUTREQ","CUTSTRATEGY","DEFAULTALG","DEGRADEFACTOR"}
  end-if
end-procedure

XADseteventcallback("guievents")

```

Mosel で XAD を使用する際の注意点:

XAD コードは Mosel モデルの任意の場所に置くことができますが、ビジュアル機能を実装するコードは、必要に応じて他のモデリング/プログラミングのステートメントと分離してください。*mmxad.dso* を使用するモデルを実行するためには、このライブラリが Xpress-MP をインストールした dso フォルダに存在し(または MOSEL_DOS 環境変数に指定され)、ライセンスファイルによって適切にライセンスを受けていなければなりません。

Chapter 2

GUI creation with XAD

XAD ユーザーインターフェースは、ユーザーによる開閉が可能なウィンドウで構成します。各ウィンドウには任意の数のリスト、ボタン、チェックボックスなどのような XAD オブジェクトを含めることができます。すべてのウィンドウは、すべての XAD オブジェクトと同じく、関連づけられた一意の整数識別子(以下 ID)を持ちます。識別子は、生成時にグラフィックオブジェクトに関連づけられた後、それぞれのオブジェクトを参照するためにプログラム/モデルで使用します。

ユーザーがXADオブジェクトを操作するとイベントが発生します。例えば、ボタンを押すと [XAD_EVENT_PRESSED](#) が発生し、リストのアイテムを選択すると [XAD_EVENT_SELECTION](#) イベントが発生します。イベントはユーザーが記述したコールバックルーチンで処理します。ユーザーインターフェースの動作は、イベントに対するアプリケーションの反応によって決まります。ユーザーインターフェースの状態を変更するためのXADステートメントを含め、任意のMoselコードをイベント処理のために記述することができます。このためMoselデータの処理が、GUIとのやりとりと同様、非常に柔軟になります。

簡単な例で、XAD ユーザーインターフェースを作成し処理する原理を示します。後でモデル中で必要となる数字を求めるウィンドウを表示する Mosel プログラムを記述します:

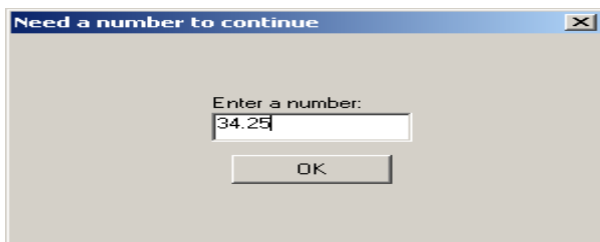


図 2.1: 簡単な入力ウィンドウの例

コードを見てみましょう:

```
model simple
uses "mumxad";

declarations
  id_win=1;id_textnumber=2;id_inputnumber=3;id_buttonok=4 !store ids for later
  N: real !the number we are seeking
end-declarations

!user interface creation
XADcreatewindow (id_win,50,50,300,200,"Need a number to continue")
XADcreatetext (id_win,id_textnumber,20,52,80,40,"Enter a number:")
XADcreateinput (id_win,id_inputnumber,100,50,100,20,"")
XADcreatebutton (id_win,id_buttonok,110,100,80,24,"OK")
```

最初にウィンドウを生成し id_win を割り当てます。次に、それぞれ異なる ID を持った説明文、テキスト、入力フィールドおよびボタンの 3 つの XAD オブジェクトを生成します。ID は短いよりは長いほうが良く、またモデルを読みやすく保守しやすいものとするため、型情報を提供するものにしてください。

次は以下のようになります:

```
!event handler
procedure guievents (id:integer, event:integer)
  if id=id_buttonok and event=XAD_EVENT_PRESSED then
    XADwindowclose(id_win)
  elif id=id_win and event=XAD_EVENT_WINDOW_CLOSED then
    N:=real(XADinputgettext(id_inputnumber))
  elif id=id_win and event=XAD_EVENT_WINDOW_OPENED then
    XADsetfocus(id_inputnumber)
  end-if
end-procedure
```

イベントハンドラーは XAD プログラムの中核をなしています。ユーザーと GUI との... やりとりは、すべてイベントハンドラーを通して反映されます。イベントハンドラーはコールバックプロシージャで、integer 型の 2 つの引数を取ります。XAD はイベントが発生すると、このプロシージャを呼びます。引数は:

id: integer イベントが発生した XAD オブジェクトの ID

event: integer イベントを示す数字(例えば[XAD_EVENT_PRESSED](#))

キーの押下やリストの選択のようなイベントには、テキスト形式の情報を伴うものがあることに注意してください。この情報が必要な場合、イベントハンドラーの中で[XADgettext](#): stringルーチンを使って取り出すことができます。

今の簡単な例の場合では、id_buttonok オブジェクトが押されるとウィンドウを閉じます。ウィンドウを閉じると、id_inputnumber 入力オブジェクトの現在のテキストを実数に変換し、N に割り当てます。3 番目に、ユーザーとのやりとりを良くするため、ウィンドウを開くと同時に入力オブジェクトにフォーカスを設定します。

最後は以下のようになります:

```
!set event handler
XADseteventcallback("guievents")

!open window - this is a "blocking" call
XADwindowopen(id_win)

!once window is closed execution continues here
writeln("N=",N)

end-model
```

イベントを処理するプロシージャ(イベントハンドラー)は、イベントハンドラーのプロシージャ名を唯一の引数として[XADseteventcallback](#)を呼び、XADに登録しなければなりません。

イベントハンドラーが登録されると、ウィンドウを開く(表示する)ことができます。ウィンドウを閉じると(OK ボタンを押す、または閉じるボタンをクリックする、または Esc を押します)、Mosel は XADwindowopen のすぐ後に続くステートメントから実行を再開します。

Chapter 3

Lifetime of XAD objects

XADオブジェクトは、モデルが終了するまで、または[XADdestroy](#) を使って破棄するまで存在します。オブジェクト固有のルーチン(本資料のリファレンス部分に記載したもの)は、オブジェクトを含むウィンドウがアクティブなときだけ(または、オブジェクトがウィンドウならウィンドウがアクティブな間だけ)、そのオブジェクトを操作できます。このため[XAD_EVENT_WINDOW_OPENED](#) イベントは特に重要です。ウィンドウがユーザーとのやりとりを開始する前に、ウィンドウのオブジェクトを操作する唯一の機会だからです。

ウィンドウには明確に区別された 2 つの動きがあります:

ウィンドウを *開く* と([XADwindowopen](#) を使います)、ウィンドウはMoselプログラムを制御します。ウィンドウを *閉じる* までは([XADwindowclose](#) を使います)、コードを実行する唯一の方法はイベントハンドラーからになります(直接、またはサブルーチン呼びます)。XADwindowopenに続くステートメントは、ユーザーがウィンドウを閉じた後にのみ実行されます。この動きは、プログラムを継続する前にユーザー入力が必要なとき、またはユーザーインターフェースがMoselプログラムを制御するようになっているときに使用することができます。このタイプのウィンドウを *モーダルダイアログ* といいます。

ウィンドウを *表示* すると([XADwindowshow](#) を使います)、ウィンドウが表れ、Moselはすぐに再開します。表示されたウィンドウは、プログラムの状態を監視したり、進行を表示したりするなど補助的なウィンドウとして使用してください。監視が(例えばMoselの実行の最後に)完了すると、[XADwindowhide](#) を使ってウィンドウを *非表示* にすることができます。Moselの実行終了後に、ウィンドウを非表示にしないと、モデルがメモリからアンロードされるまでウィンドウは存在します(これは、Xpress-IVEでは、モデルのコンパイルまたは実行時にのみ生じます)。このタイプのウィンドウを *モードレスダイアログ* といいます。

2 つの動きは正しく選択することが重要です。違いを十分理解するまで試験し、たいていの目的に対しては、イベントの発生によってプログラムの実行を制御することができるようにするために、ウィンドウを開くべきである点に注意してください。

Chapter 4

Events

ユーザーが、ウィンドウまたはウィンドウ中のオブジェクト上で操作をしたとき、その行為はイベントを通して伝えられます。イベントハンドラーのコールバックプロシージャは、そのようなイベントに応答するために使用することができます。例えば、ユーザーがボタンを押したとき、イベントハンドラーのコールバックプロシージャは、2つの引数とともに呼ばれます。1つはイベントのトリガーとなったボタンのIDを表す整数で、もう1つは[XAD_EVENT_PRESSED](#) イベントコードを表す整数です。また、ユーザーがウィンドウの縁をドラッグしてウィンドウのサイズを変更したとき、イベントは、ウィンドウIDおよび[XAD_EVENT_WINDOW_RESIZED](#) を引数とするイベントハンドラーのコールバックの呼出しとして伝えられます。

Chapter 5

XAD objects reference

XAD は、以下のグラフィックオブジェクトを扱う機能を備えています:

- ウィンドウ([Window](#))
- テキスト([Text](#))
- ボタン([Simple button](#))
- 入力([Input](#))
- エディタ([GUI editor](#))
- チェックボタン([Check button](#))
- ラジオボタン([Radio button](#))
- オブジェクトグループ([Group of XAD objects](#))
- リスト([List of items](#))
- プログレスバー([Progress bar](#))
- タブ([Tab selectors](#))
- マルチリスト([List object with multiple columns](#))
- キャンバス([Canvas](#))
- ブラウザ([Browser](#))

5.1 Window



固有サブルーチン

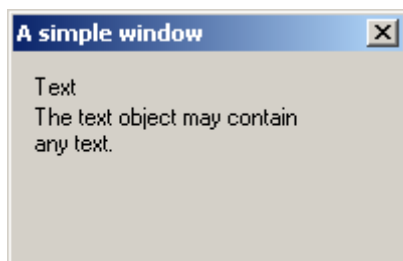
[XADcreatewindow](#), [XADwindowopen](#), [XADwindowclose](#), [XADwindowshow](#),
[XADwindowhide](#), [XADwindowkeep](#), [XADwindowsettimer](#)

固有イベント

[XAD_EVENT_TIMER](#), [XAD_EVENT_WINDOW_CLOSED](#),
[XAD_EVENT_WINDOW_CLOSING](#), [XAD_EVENT_WINDOW_HIDDEN](#),
[XAD_EVENT_WINDOW_MOVED](#), [XAD_EVENT_WINDOW_OPENED](#),

[XAD_EVENT_WINDOW_RESIZED](#), [XAD_EVENT_WINDOW_SHOWN](#)

5.2 Text



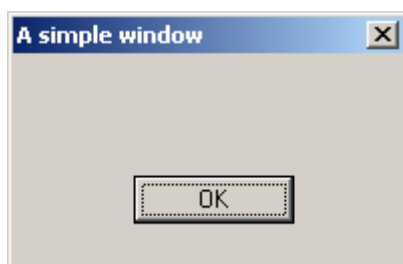
固有サブルーチン

[XADcreatetext](#), [XADtextaddtext](#), [XADtextsettext](#), [XADtextgettext](#)

固有イベント

なし

5.3 Simple button



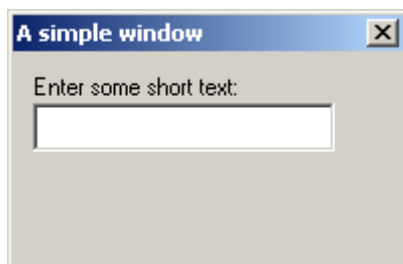
固有サブルーチン

[XADcreatebutton](#)

固有イベント

[XAD_EVENT_PRESSED](#)

5.4 Input



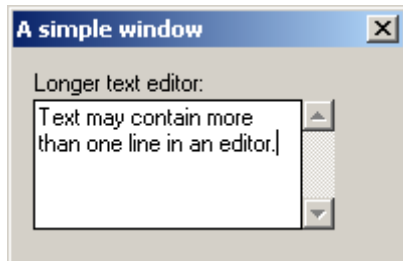
固有サブルーチン

[XADcreateinput](#), [XADinputsettext](#), [XADinputgettext](#)

固有イベント

[XAD_EVENT_CHANGED](#)

5.5 GUI editor



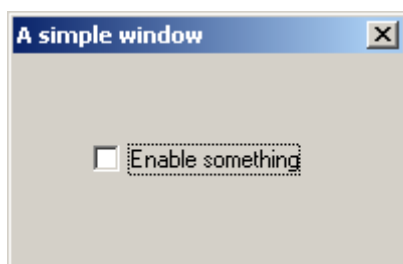
固有サブルーチン

[XADcreateeditor](#), [XADeditoraddtext](#), [XADeditorload](#), [XADeditorsave](#),
[XADeditorsettext](#), [XADeditorgettext](#)

固有イベント

[XAD_EVENT_CHANGED](#)

5.6 Check button



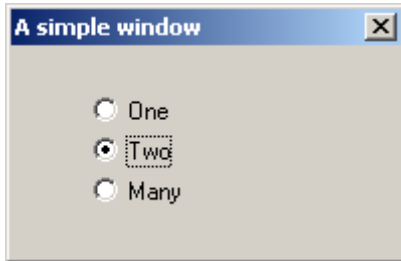
固有サブルーチン

[XADcreatecheck](#), [XADchecksetstate](#), [XADcheckgetstate](#)

固有イベント

[XAD_EVENT_PRESSED](#)

5.7 Radio button



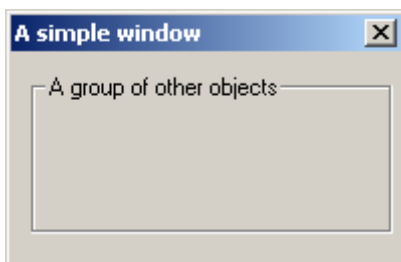
固有サブルーチン

[XADcreateradio](#), [XADradiosetstate](#), [XADradiogetstate](#)

固有イベント

[XAD_EVENT_PRESSED](#)

5.8 Group of XAD objects



固有サブルーチン

[XADcreategroup](#)

固有イベント

なし

5.9 List of items



固有サブルーチン

[XADcreatelists](#), [XADlistadd](#), [XADlistgetsel](#), [XADlistshow](#)

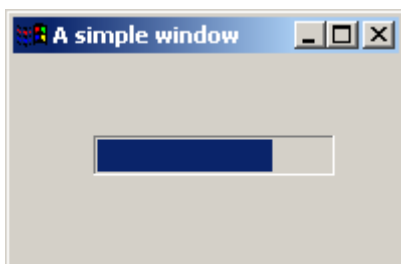
固有イベント

[XAD_EVENT_SELECTION](#)

注意

リストオブジェクトでは、ユーザーが選択を変更すると[XAD_EVENT_SELECTION](#) イベントが発生します。どのアイテムが選択されたかを調べるためには、[XADgeteventtext](#): string 関数を呼んでください。

5.10 Progress bar



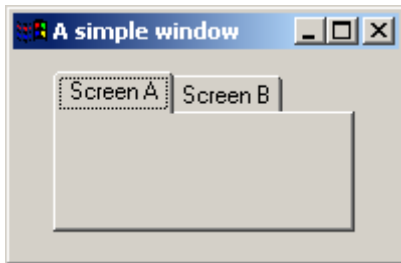
固有サブルーチン

[XADcreateprogress](#), [XADprogressset](#)

固有イベント

なし

5.11 Tab selectors



固有サブルーチン

[XADcreatetab](#)

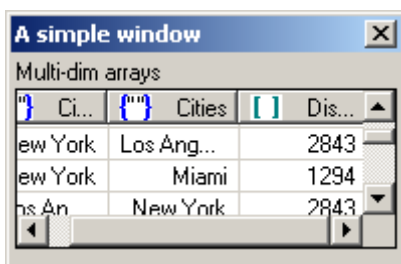
固有イベント

[XAD_EVENT_SELECTION](#)

注意

タブオブジェクトでは、ユーザーが選択を変更すると[XAD_EVENT_SELECTION](#) イベントが発生します。どのタブが選択されたかを調べるためには、[XADgeteventtext](#):string 関数を呼んでください。

5.12 List object with multiple columns



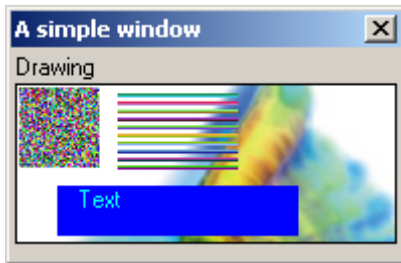
固有サブルーチン

[XADcreatemultilist](#), [XADmultilistshow](#), [XADmultilistsetsize](#),
[XADmultilistsetcolname](#), [XADmultilistsettext](#)

固有イベント

なし

5.13 Canvas



固有サブルーチン

[XADcreatecanvas](#), [XADcanvasdrawbox](#), [XADcanvasdrawellipse](#),
[XADcanvaserase](#), [XADcanvasrefresh](#), [XADcanvasdrawimage](#),
[XADcanvassaveimage](#), [XADcanvasdrawline](#), [XADcanvasdrawpoint](#),
[XADcanvasdrawrectangle](#), [XADcanvasdrawtext](#), [XADcanvasmap](#),
[XADcanvasunmap](#), [XADcolor](#)

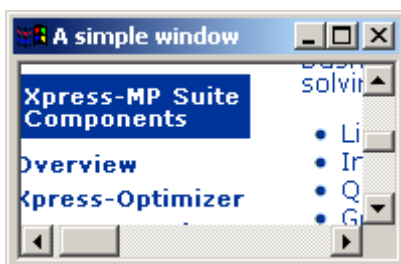
固有イベント

なし

注意

色を扱う際、XAD は次の色定数を認識します: XAD_BLACK, XAD_BLUE, XAD_CYAN, XAD_GREEN, XAD_MAGENTA, XAD_ORANGE, XAD_RED, XAD_WHITE, XAD_YELLOW

5.14 Browser



固有サブルーチン

[XADcreatebrowser](#), [XADbrowsergoto](#)

固有イベント

なし

5.15 Subroutines specific to objects

XADbrowsergoto	指定された URL をブラウザで開きます。	p.73
XADcanvasdrawbox	キャンバスにボックスを描画します。	p.58
XADcanvasdrawellipse	キャンバスに楕円を描画します。	p.59
XADcanvasdrawimage	ファイルの画像を描画します。	p.62
XADcanvasdrawline	キャンバスに線を描画します。	p.64
XADcanvasdrawpoint	キャンバスに点を描画します。	p.65
XADcanvasdrawrectangle	キャンバスに長方形を描画します。	p.66
XADcanvasdrawtext	キャンバスにテキストを描画します。	p.67
XADcanvaserase	キャンバスを消去します。	p.60
XADcanvasmap	キャンバスの座標空間をマッピングします。	p.68
XADcanvasrefresh	キャンバスを再描画します。	p.61
XADcanvassaveimage	画像をファイルに保存します。	p.63
XADcanvasunmap	キャンバスのデフォルトのマッピングに戻します。	p.70
XADcheckgetstate	チェックボタンの状態を取得します。	p.40
XADchecksetstate	チェックボタンの状態を設定します。	p.39
XADcolor	明度を生成します。	p.71
XADcreatebrowser	ブラウザを生成します。	p.72
XADcreatebutton	ボタンオブジェクトを生成します。	p.28
XADcreatecanvas	キャンバスオブジェクトを生成します。	p.57
XADcreatecheck	チェックボタンを生成します。	p.38

XADcreateeditor	エディタオブジェクトを生成します。	p.32
XADcreategroup	グループオブジェクトを生成します。	p.44
XADcreateinput	入力オブジェクトを生成します。	p.29
XADcreatelist	リストオブジェクトを生成します。	p.45
XADcreatemultilist	マルチリストオブジェクトを生成します。	p.52
XADcreateprogress	プログレスバーを生成します。	p.49
XADcreateradio	ラジオボタンを生成します。	p.41
XADcreatetab	タブセレクタオブジェクトを生成します。	p.51
XADcreatetext	テキストオブジェクトを生成します。	p.24
XADcreatewindow	ウィンドウを生成します。	p.17
XADeditoraddtext	エディタにテキストを追加します。	p.33
XADeditorgettext	エディタからテキストを取得します。	p.37
XADeditorload	エディタにファイルをロードします。	p.34
XADeditorsave	エディタの内容をファイルに保存します。	p.35
XADeditorsettext	エディタのテキストを設定します。	p.36
XADinputgettext	入力オブジェクトのテキストを取得します。	p.31
XADinputsettext	入力オブジェクトのテキストを設定します。	p.30
XADlistadd	リストにアイテムを追加します。	p.46
XADlistgetsel	リストから選択されたアイテムを取得します。	p.47
XADlistshow	リストを表示します。	p.48
XADmultilistsetcolname	マルチリストのカラム名を設定します。	p.55

<u>XADmultilistsetsize</u>	マルチリストのサイズを(再)設定します。	p.54
<u>XADmultilistsettext</u>	マルチリストのアイテムを設定します。	p.56
<u>XADmultilistshow</u>	マルチリストを表示します。	p.53
<u>XADprogressset</u>	進行状況を設定します。	p.50
<u>XADradiogetstate</u>	ラジオボタンの状態を取得します。	p.43
<u>XADradiosetstate</u>	ラジオボタンの状態を設定します。	p.42
<u>XADtextaddtext</u>	テキストオブジェクトにテキストを追加します。	p.25
<u>XADtextgettext</u>	テキストオブジェクトのテキストを取得します。	p.27
<u>XADtextsettext</u>	テキストオブジェクトのテキストを設定します。	p.26
<u>XADwindowclose</u>	ウィンドウを閉じます。	p.19
<u>XADwindowhide</u>	ウィンドウを非表示にします。	p.21
<u>XADwindowkeep</u>	ウィンドウの状態を保ちます。	p.22
<u>XADwindowopen</u>	ウィンドウを開きます。	p.18
<u>XADwindowsettimer</u>	タイマーを(再)設定します。	p.23
<u>XADwindowshow</u>	ウィンドウを表示します。	p.20

XADcreatewindow

Purpose

Create a window

Synopsis

procedure XADcreatewindow(id:integer, x:integer, y:integer, w:integer, h:integer, name:string)

Arguments

id Arbitrary unique id of the window

x x coordinate (from left of screen) of the window in pixels

y y coordinate (from top of screen) of the window in pixels

w Width of the window in pixels

h Height of the window in pixels

name The title of the window

Related topics

[XADwindowopen](#), [XADwindowclose](#), [XADwindowshow](#), [XADwindowhide](#),
[XADwindowkeep](#), [XADwindowsettimer](#)

XADwindowopen

Purpose

Open a window.

Synopsis

procedure XADwindowopen(id:integer)

Arguments

 id Object identifier

Further information

Displays the window and surrenders execution control to its event handler callback.

Related topics

[XADwindowclose](#)

XADwindowclose

Purpose

Close a window.

Synopsis

procedure XADwindowclose(id: integer)

Argument

 id Object identifier

Further information

Destroys an *opened* window.

Related topics

[XADwindowopen](#)

XADwindowshow

Purpose

Display a window.

Synopsis

procedure XADwindowshow(id: integer)

Argument

 id Object identifier

Further information

Displays the window and execution resumes immediately.

Related topics

[XADwindowhide](#)

XADwindowhide

Purpose

Hide a window.

Synopsis

procedure XADwindowhide(id: integer)

Argument

 id Object identifier

Further information

Destroys a *shown* window.

Related topics

[XADwindowshow](#)

XADwindowkeep

Purpose

Keep a window.

Synopsis

procedure XADwindowkeep

Further information

Should be used only when dealing with the [XAD_EVENT_WINDOW_CLOSING](#) event, if the intent is to cancel the closing of the window. For example, if an exit confirmation dialog is shown and the user changes his mind, call this procedure to prevent the window from closing.

Related topics

[XADwindowhide](#)

XADwindowsettimer

Purpose

(Re)set a timer.

Synopsis

procedure XADwindowsettimer(id: integer, msec: integer)

Arguments

id Object identifier

msec Timer interval in milliseconds

Further information

Sets or resets a timer that generates [XAD_EVENT_TIMER](#) events. If *msec* is a positive integer, the event will be generated every *msec* milliseconds. If *msec* is 0 or negative, the timer is reset (timer events are no longer generated)

XADcreatetext

Purpose

Create a text object.

Synopsis

procedure XADcreatetext(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer, name:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

text Contents of the text object (use `¥r¥n` to break lines)

Related topics

[XADtextaddtext](#), [XADtextgettext](#), [XADtextsettext](#)

XADtextaddtext

Purpose

Add text to a text object.

Synopsis

procedure XADtextaddtext(id: integer, text: string)

Arguments

 id Object identifier

 text Text to be added

Further information

Appends the given *text* to the object.

Related topics

[XADtextgettext](#), [XADtextsettext](#)

XADtextsettext

Purpose

Set text of a text object.

Synopsis

procedure XADtextsettext(id: integer, text: string)

Arguments

 id Object identifier

 text New text

Further information

Replaces the text of the object with the new *text*.

Related topics

[XADtextaddtext](#), [XADtextgettext](#)

XADtextgettext

Purpose

Get the text of a text object.

Synopsis

```
function XADtextgettext(id: integer): string
```

Argument

 id Object identifier

Return value

Text of the object.

Further information

Obtains the text currently shown by the object.

Related topics

[XADtextaddtext](#), [XADtextsettext](#)

XADcreatebutton

Purpose

Create a button object.

Synopsis

procedure XADcreatebutton(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer, name:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

name The name of the button

XADcreateinput

Purpose

Create an input object.

Synopsis

procedure XADcreateinput(wid: integer, id: integer, x: integer, y: integer, w: integer, h: integer, text: string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

text Initial content of the input object

Related topics

[XADinputgettext](#), [XADinputsettext](#)

XADinputsettext

Purpose

Set the text of an input object.

Synopsis

procedure XADinputsettext(id: integer, text: string)

Arguments

 id Object identifier

 text New text

Further information

Replaces the *text* of the object with the new text.

Related topics

[XADinputgettext](#)

XADinputgettext

Purpose

Get the text of an input object.

Synopsis

```
function XADinputgettext(id: integer): string
```

Argument

 id Object identifier

Return value

Text of the object.

Further information

Obtains the text currently shown by the object.

Related topics

[XADinputsettext](#)

XADcreateeditor

Purpose

Create an editor object.

Synopsis

procedure XADcreateeditor(wid:integer, id:integer, x:integer, y:integer, w:integer, h:integer, text:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

text Contents of the editor object (use `¥r¥n` to break lines)

Related topics

[XADeditoraddtext](#), [XADeditorload](#), [XADeditorsave](#), [XADeditorsettext](#),
[XADeditorgettext](#)

XADeditoraddtext

Purpose

Add text to an editor.

Synopsis

procedure XADeditoraddtext(id: integer, text: string)

Arguments

 id Object identifier

 text Text to be added

Further information

Appends the given *text* to the object.

Related topics

[XADeditorsettext](#), [XADeditorgettext](#)

XADeditorload

Purpose

Load a file into an editor.

Synopsis

procedure XADeditorload(id: integer, file: string)

Arguments

id Object identifier

file File name

Further information

Load the specified *file* into the editor, replacing its previous content.

Related topics

[XADeditorsave](#)

XADeditorsave

Purpose

Save editor contents into a file.

Synopsis

procedure XADeditorsave(id:integer, file:string)

Arguments

id Object identifier

file File name

Further information

Saves the contents of the editor to the specified *file*.

Related topics

[XADeditorload](#)

XADeditorsettext

Purpose

Set the text of an editor.

Synopsis

procedure XADeditorsettext(id: integer, text: string)

Arguments

id Object identifier

text New text

Further information

Replaces the text of the object with the new *text*.

Related topics

[XADeditoraddtext](#), [XADeditorgettext](#)

XADeditorgettext

Purpose

Get the text from an editor.

Synopsis

```
function XADeditorgettext(id: integer): string
```

Argument

id Object identifier

Return value

Text of the object.

Further information

Obtains the text currently shown by the object.

Related topics

[XADeditoraddtext](#), [XADeditorsettext](#),

XADcreatecheck

Purpose

Create a check button.

Synopsis

procedure XADcreatecheck(wid:integer, id:integer, x:integer, y:integer, w:integer, h:integer, name:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

name Description of the check object

Related topics

[XADchecksetstate](#), [XADcheckgetstate](#)

XADchecksetstate

Purpose

Set the state of a check button.

Synopsis

procedure XADchecksetstate(id:integer, state:boolean)

Arguments

id Object identifier

state Button state.

 true checked

 false unchecked

Further information

Sets the *state* of the check to checked or unchecked.

Related topics

[XADcheckgetstate](#)

XADcheckgetstate

Purpose

Get the state of a check button.

Synopsis

```
function XADcheckgetstate(id: integer): boolean
```

Argument

id Object identifier

Return value

true Button is checked

false Button is unchecked

Further information

Obtains the state of the check (`true` for checked, `false` for unchecked).

Related topics

[XADchecksetstate](#)

XADcreateradio

Purpose

Create a radio button.

Synopsis

procedure XADcreateradio(wid: integer, id: integer, x: integer, y: integer, w: integer, h: integer, name: string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

name Name of the radio button object

Related topics

[XADradiosetstate](#), [XADradiogetstate](#)

XADradiosetstate

Purpose

Get the state of a radio button.

Synopsis

procedure XADradiosetstate(id: integer, state: boolean)

Arguments

id Object identifier

state Button state.

 true selected

 false unselected

Further information

Sets the *state* of the radio to selected (true) or unselected (false).

Related topics

[XADradiogetstate](#)

XADradiogetstate

Purpose

Get the state of a radio button.

Synopsis

```
function XADradiogetstate(id: integer): boolean
```

Arguments

id Object identifier

Return value

true Button is selected

false Button is unselected

Further information

Obtains the state of the radio (true for selected, false for unselected).

Related topics

[XADradiosetstate](#)

XADcreategroup

Purpose

Create a group object.

Synopsis

procedure XADcreategroup(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer, name:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

name Description of the group

XADcreatelist

Purpose

Create a list object.

Synopsis

procedure XADcreatelist(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer, name:string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

name String containing comma-separated list items

Related topics

[XADlistadd](#), [XADlistgetsel](#), [XADlistshow](#)

XADlistadd

Purpose

Add an item to a list

Synopsis

procedure XADlistadd(id: integer, item: string)

Arguments

 id Object identifier

 item Item to add

Further information

Appends the *item* to the list.

Related topics

[XADlistgetsel](#), [XADlistshow](#)

XADlistgetsel

Purpose

Get the selected item from a list.

Synopsis

```
function XADlistgetsel(id: integer): string
```

Arguments

id Object identifier

Return value

Selected item.

Further information

Obtains the item currently selected.

Related topics

[XADlistadd](#), [XADlistshow](#)

XADlistshow

Purpose

Show a list.

Synopsis

procedure XADlistshow(id:integer, items:set)

Arguments

 id Object identifier

 items Items to display

Further information

Fills the list with the contents of the given set of *items*. Mosel will try to infer the type of the set and display accordingly. For example showing a *set of mpvar* will display the solution values and reduced costs for each item.

Related topics

[XADlistadd](#), [XADlistgetsel](#)

XADcreateprogress

Purpose

Create a progress bar.

Synopsis

procedure XADcreateprogress(wid:integer, id:integer, x:integer, y:integer, w:integer, h:integer)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

Related topics

[XADprogressset](#)

XADprogressset

Purpose

Set the progress state.

Synopsis

```
procedure XADprogressset(id: integer, minval: real, maxval: real, current: real)
```

Arguments

id Object identifier

minval Lower bound

maxval Upper bound

current Current state

Further information

The progress will show how far *current* is between *minval* and *maxval*.

XADcreatetab

Purpose

Create a tab selector object.

Synopsis

procedure XADcreatetab(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer, tabs:set of string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

tabs Set of strings containing the possible choices

XADcreatemultilist

Purpose

Create a multilist object.

Synopsis

procedure XADcreatemultilist(wid:integer, id:integer, name:string, x:integer, y:integer, w:integer, h:integer)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

Related topics

[XADmultilistshow](#),

[XADmultilistsetsize](#),

[XADmultilistsetcolname](#),

[XADmultilistsettext](#)

XADmultilistshow

Purpose

Show a multilist

Synopsis

procedure XADmultilistshow(id: integer, items: array)

Arguments

id Object identifier

items Items to display as multilist.

Further information

Clears the content of the multilist and fills it with the given array of *items* (integers, reals, strings, booleans, mpvars or linctrs).

Related topics

[XADmultilistsetsize](#), [XADmultilistsetcolname](#), [XADmultilistsettext](#)

XADmultilistsetsize

Purpose

(Re)set the size of a multilist.

Synopsis

procedure XADmultilistsetsize(id:integer, rows:integer, columns:integer)

Arguments

id	Object identifier
rows	Number of rows
columns	Number of columns

Further information

Clears the content of the multilist and prepares to hold the given number of *rows* and *columns*.

Related topics

[XADmultilistsetcolname](#), [XADmultilistsettext](#)

XADmultilistsetcolname

Purpose

Set multilist column names.

Synopsis

procedure XADmultilistsetcolname(id: integer, columns: integer, name: string)

Arguments

id Object identifier

column Column index

name Column name

Further information

The header of the *column* will display the new *name*.

Related topics

[XADmultilistsetsize](#), [XADmultilistsetcolname](#)

XADmultilistsettext

Purpose

Set a multilist item.

Synopsis

procedure XADmultilistsettext(id:integer, row:integer, column:integer,
text:string)

Arguments

id	Object identifier
column	Column index
name	Column name
text	New text of the item

Further information

The item at the given *row* and *column* will hold the new *text*.

Related topics

[XADmultilistsetsize](#), [XADmultilistsetcolname](#)

XADcreatecanvas

Purpose

Create a canvas object.

Synopsis

procedure XADcreatecanvas(wid:integer, id:integer, x:integer, y:integer,
w:integer, h:integer)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object in pixels

y y coordinate (from top of window) of the object in pixels

w Width of the object in pixels

h Height of the object in pixels

Related topics

[XADcanvasdrawbox](#), [XADcanvasdrawellipse](#), [XADcanvaserase](#),
[XADcanvasrefresh](#), [XADcanvasdrawimage](#), [XADcanvassaveimage](#),
[XADcanvasdrawline](#), [XADcanvasdrawpoint](#), [XADcanvasdrawrectangle](#),
[XADcanvasdrawtext](#), [XADcanvasmap](#), [XADcanvasunmap](#), [XADcolor](#)

XADcanvasdrawbox

Purpose

Draw a box on a canvas.

Synopsis

procedure XADcanvasdrawbox(id:integer, x:real, y:real, w:real, h:real,
color1:integer, color2:integer)

Arguments

id	Canvas identifier
x	x coordinate (from left of canvas)
y	y coordinate (from top of canvas)
w	width of box
h	height of box
color1	Border color
color2	Fill color

Further information

Draws a solid rectangle of color *color2* with a one-pixel margin of color *color1* of width *w* and height *h* at coordinates *(x,y)*.

Related topics

[XADcanvasdrawline](#), [XADcanvasdrawpoint](#), [XADcanvasdrawrectangle](#),
[XADcanvasdrawtext](#), [XADcolor](#)

XADcanvasdrawellipse

Purpose

Draw an ellipse on a canvas.

Synopsis

procedure XADcanvasdrawellipse(id:integer, x:real, y:real, w:real, h:real, color1:integer, color2:integer)

Arguments

id	Canvas identifier
x	x coordinate (from left of canvas)
y	y coordinate (from top of canvas)
w	width of ellipse
h	height of ellipse
color1	Border color
color2	Fill color

Further information

Draws a solid ellipse of color *color2* with a one-pixel margin of color *color1* of width *w* and height *h* at coordinates (x,y) .

Related topics

[XADcanvasdrawline](#), [XADcanvasdrawpoint](#), [XADcanvasdrawrectangle](#), [XADcanvasdrawtext](#), [XADcolor](#)

XADcanvaserase

Purpose

Erase a canvas.

Synopsis

procedure XADcanvaserase(id:integer, color:integer)

Arguments

 id Object identifier

 color Color selection

Further information

Clears the contents of the canvas using the specified *color*

Related topics

[XADcanvasrefresh](#)

XADcanvasrefresh

Purpose

Redraw a canvas.

Synopsis

procedure XADcanvasrefresh(id: integer)

Argument

 id Object identifier

Further information

Updates the visual content of the canvas. Simply drawing on a canvas does not update its appearance (to save time). Only call XADcanvasrefresh when necessary.

Related topics

[XADcanvasease](#)

XADcanvasdrawimage

Purpose

Draw an image from file.

Synopsis

```
procedure XADcanvasdrawimage(id:integer, x:real, y:real, file: string)
```

```
procedure XADcanvasdrawimage(id:integer, x:real, y:real, w:integer,  
h:integer, file: string)
```

Arguments

id Object identifier
x x coordinate
y y coordinate
w Width
h Height
file File name

Further information

Draws the image from the given *file* at coordinates (x,y) . The file can be of type .bmp, .jpg, .gif, or .png. If height and width are given, then the image is rescaled to fit in a rectangle of width w and height h . If the string is "xading:imgname", a previously saved *imgname* (see above) will be drawn. If the string is "xading:imgname", a previously saved *imgname* will be drawn.

Related topics

[XADcanvassaveimage](#)

XADcanvassaveimage

Purpose

Save an image to a file.

Synopsis

procedure XADcanvassaveimage(id:integer, x:real, y:real, w:integer, h:integer, imgname: string)

Arguments

id	Object identifier
x	x coordinate
y	y coordinate
w	Width
h	Height
imgname	File name

Further information

Saves what is currently drawn at x,y,w,h under the identifier *imgname*. *imgname* can then be drawn anywhere else using [XADcanvasdrawimage](#).

Related topics

[XADcanvasdrawimage](#)

XADcanvasdrawline

Purpose

Draw a line on a canvas.

Synopsis

```
procedure XADcanvasdrawline(id:integer, x1:real, y1:real, x2:real, y2:real,  
color:integer)
```

```
procedure XADcanvasdrawline(id:integer, x1:real, y1:real, x2:real, y2:real,  
color:integer, width: integer)
```

Arguments

id	Object identifier
x1	x start coordinate
y1	y start coordinate
x2	x end coordinate
y2	y end coordinate
color	Color
width	Width

Further information

Draws a line *width*-pixels wide from $(x1,y1)$ to $(x2,y2)$ with the given *color*.
The default width is one pixel.

Related topics

[XADcanvasdrawbox](#), [XADcanvasdrawpoint](#), [XADcanvasdrawrectangle](#),
[XADcanvasdrawtext](#), [XADcolor](#)

XADcanvasdrawpoint

Purpose

Draw a point on a canvas.

Synopsis

procedure XADcanvasdrawpoint>(id: integer, x: real, y: real, color: integer)

Arguments

id Object identifier

x x coordinate

y y coordinate

color Color

Further information

Draws a pixel at (x,y) with the given *color*.

Related topics

[XADcanvasdrawbox](#), [XADcanvasdrawline](#), [XADcanvasdrawrectangle](#),
[XADcanvasdrawtext](#), [XADcolor](#)

XADcanvasdrawrectangle

Purpose

Draw a rectangle on a canvas.

Synopsis

procedure XADcanvasdrawrectangle>(id:integer, x:real, y:real, w:real, h:real, color:integer)

Arguments

id	Object identifier
x	x coordinate
y	y coordinate
w	Width
h	Height
color	Color

Further information

Draws a solid rectangle of width w and height h at coordinates (x,y) with the given *color*.

Related topics

[XADcanvasdrawbox](#), [XADcanvasdrawline](#), [XADcanvasdrawpoint](#),
[XADcanvasdrawtext](#), [XADcolor](#)

XADcanvasdrawtext

Purpose

Draw text on a canvas.

Synopsis

```
procedure XADcanvasdrawtext(id:integer, x:real, y:real, text:string,  
color:integer)
```

```
procedure XADcanvasdrawtext(id:integer, x:real, y:real, text:string,  
color:integer, fontsize:integer, alignment:integer, fontname:string)
```

Arguments

id	Object identifier
x	x coordinate
y	y coordinate
text	Text
color	Color
fontsize	Font size
alignment	Vertical and horizontal alignment
fontname	Font name

Further information

Draws *text* with font *fontname* of size *fontsize* at coordinates (x,y) with the given *color* and *alignment*. If font name and size are not given then the output uses Arial 10pt. The alignment is the sum of one of XAD_CENTERH, XAD_LEFT, and XAD_RIGHT (horizontal alignment) plus one of XAD_CENTERV, XAD_TOP, and XAD_BOTTOM (vertical alignment). XAD_DEFAULT can be used instead to specify upper left alignment.

Related topics

[XADcanvasdrawbox](#), [XADcanvasdrawline](#), [XADcanvasdrawpoint](#),
[XADcanvasdrawrectangle](#), [XADcolor](#)

XADcanvasmap

Purpose

Map the coordinate space of a canvas.

Synopsis

```
procedure XADcanvasmap(id:integer, x1:real, y1:real, x2:real, y2:real,  
x1new:real, y1new:real, x2new:real, y2new:real)
```

Arguments

id	Object identifier
x1	x start coordinate
y1	y start coordinate
x2	x end coordinate
y2	y end coordinate
x1new	New x start coordinate
y1new	New y start coordinate
x2new	New x end coordinate
y2new	New y end coordinate

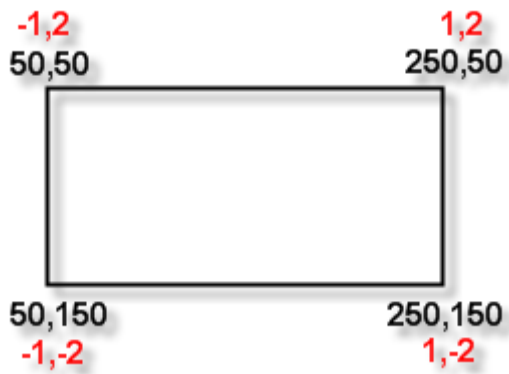
Further information

Transforms the coordinate space of the canvas. Normally, (0,0) represents the upper left corner of the canvas and widths and heights are measured in pixels. By remapping the coordinate space, transformations no longer need to be applied to coordinates when drawing. For example, if the canvas has a width of 300 and a height of 200 and if we want to draw a graph from -1 to +1 on the x axis and -2 to +2 on the y axis in a region of the original canvas, we could write, say:

```
XADcanvasmap(id_canvas, 50, 150, 250, 50, -1, -2, 1, 2)
```

before plotting the points on the graph. Note that this transformation reverts the direction of the y axis and plots everything between (-1,1) and (-2,2) in the portion of the canvas between pixel coordinates (50,50) and (250,150).

The following figure clarifies the effect of the XADcanvasmap call:



After calling this procedure, coordinates should be given in the *red* intervals. XAD will map them correctly onto the canvas, based on the *black* intervals.

Related topics

[XADcanvasmap](#)

XADcanvasunmap

Purpose

Revert to default mapping for a canvas.

Synopsis

procedure XADcanvasunmap(id: integer)

Argument

 id Object identifier

Further information

Reverts to default mapping of coordinates

Related topics

[XADcanvasunmap](#)

XADcolor

Purpose

Create a color value.

Synopsis

```
function XADcolor(red:real, green:real, blue:real):integer
```

Arguments

red Intensity of red (between 0 and 255)

green Intensity of green (between 0 and 255)

blue Intensity of blue (between 0 and 255)

Return value

Color value.

Further information

Creates a color value based on intensities of *red*, *green* and *blue*.

Related topics

See Section [Canvas](#) for a list of predefined color constants.

XADcreatebrowser

Purpose

Create a browser.

Synopsis

procedure XADcreatebrowser(wid: integer, id: integer, x: integer, y: integer, w: integer, h: integer, url: string)

Arguments

wid id of the window containing this object

id id of this object

x x coordinate (from left of window) of the object
in pixels

y y coordinate (from top of window) of the object
in pixels

w Width of the object in pixels

h Height of the object in pixels

url URL to open when the object is created

Related topics

[XADbrowsergoto](#)

XADbrowsergoto

Purpose

Open the given URL in the browser.

Synopsis

procedure XADbrowsergoto(id:integer, url:string)

Arguments

 id Object identifier

 url The URL to visit.

Further information

Visits the given *url*.

Related topics

[XADcreatebrowser](#)

5.16 Events specific to objects

XAD_EVENT_CHANGED	入力またはエディタが変更されたときのイベント。	p.76
XAD_EVENT_PRESSED	ボタンが押されたときのイベント。	p.76
XAD_EVENT_SELECTION	選択のイベント。	p.76
XAD_EVENT_TIMER	タイマーのイベント。	p.74
XAD_EVENT_WINDOW_CLOSED	ウィンドウを閉じたときのイベント。	p.74
XAD_EVENT_WINDOW_CLOSING	ウィンドウを閉じているときのイベント。	p.75
XAD_EVENT_WINDOW_HIDDEN	ウィンドウを非表示にしたときのイベント。	p.75
XAD_EVENT_WINDOW_MOVED	ウィンドウを移動したときのイベント。	p.75
XAD_EVENT_WINDOW_OPENED	ウィンドウを開いたときのイベント。	p.75
XAD_EVENT_WINDOW_RESIZED	ウィンドウのサイズを変更したときのイベント。	p.75
XAD_EVENT_WINDOW_SHOWN	ウィンドウを表示したときのイベント。	p.75

XAD_EVENT_TIMER

Description Timer event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_CLOSED

Description Window closed event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_CLOSING

Description Window closing event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_HIDDEN

Description Window hidden event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_MOVED

Description Window moved event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_OPENED

Description Window opened event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_RESIZED

Description Window resized event.

Notes Event associated with *window* objects.

XAD_EVENT_WINDOW_SHOWN

Description Window shown event.

Notes Event associated with *window* objects.

XAD_EVENT_PRESSED

Description Button pressed event.

Notes Event associated with simple *button* objects, *check* buttons, and *radio* buttons.

XAD_EVENT_CHANGED

Description Input or editor changed event.

Notes Event associated with *input* and *editor* objects.

XAD_EVENT_SELECTION

Description Selection event.

Notes Event associated with *list* and *tab* objects.

Chapter6

Generic routines

XADdestroy	Delete an object.	p.78
XADenable	Enable/disable user interaction.	p.79
XADgeteventtext	Retrieve the message associated with an event.	p.89
XADgeth	Get the height of an object.	p.83
XADgetw	Get the width of an object.	p.82
XADgetx	Get the x coordinate of an object.	p.80
XADgety	Get the y coordinate of an object.	p.81
XADrefresh	Refresh an object.	p.84
XADsetfocus	Focus on an object	p.85
XADsetpos	Reposition an object.	p.86
XADsettext	(Re)set the textual information of an object.	p.87
XADsetvisible	Making an object visible/hidden.	p.88

XADdestroy

Purpose

Delete an object.

Synopsis

procedure XADdestroy(id: integer)

Argument

 id Object identifier

Further information

When an object is no longer needed, it can be destroyed. Once destroyed, its *id* can no longer be used. However, a new object can be created with the same *id*.

XADenable

Purpose

Enable/disable user interaction.

Synopsis

```
procedure XADenable(id: integer, state: boolean)
```

Arguments

id Object identifier

state true enable user interaction with the object

 false disable user interaction with the object

Further information

Used to enable/disable user interaction with the object. For example, a *Save* button should probably be disabled if there is nothing to save.

XADgetx

Purpose

Get the x coordinate of an object.

Synopsis

```
function XADgetx(id: integer): integer
```

Argument

id Object identifier

Return value

x coordinate value of the object.

Further information

This function returns the x coordinate (in pixels) of the object *id* relative to the upper left corner of the window containing it. If *id* refers to a *window*, it returns the coordinate relative to the screen.

Related topics

[XADgety](#), [XADgeth](#), [XADgetw](#)

XADgety

Purpose

Get the y coordinate of an object.

Synopsis

```
function XADgety(id: integer): integer
```

Argument

id Object identifier

Return value

y coordinate value of the object.

Further information

This function returns the y coordinate (in pixels) of the object *id* relative to the upper left corner of the window containing it. If *id* refers to a *window*, it returns the coordinate relative to the screen.

Related topics

[XADgetx](#), [XADgeth](#), [XADgetw](#)

XADgetw

Purpose

Get the width of an object.

Synopsis

```
function XADgetw(id:integer):integer
```

Argument

 id Object identifier

Return value

Width of the object.

Further information

This function returns the width (in pixels) of the object.

Related topics

[XADgeth](#), [XADgetx](#), [XADgety](#)

XADgeth

Purpose

Get the height of an object.

Synopsis

```
function XADgeth(id: integer): integer
```

Argument

id Object identifier

Return value

Height of the object.

Further information

This function returns the height (in pixels) of the object.

Related topics

[XADgetw](#), [XADgetx](#), [XADgety](#)

XADrefresh

Purpose

Refresh an object.

Synopsis

procedure XADrefresh(id: integer)

Argument

 id Object identifier

Further information

If an object may not be painted correctly due to a complex layout operation, use this routine to update its appearance.

XADsetfocus

Purpose

Focus on an object

Synopsis

procedure XADsetfocus(id:integer)

Argument

 id Object identifier

Further information

The target object will receive the keyboard focus.

XADsetpos

Purpose

Reposition an object.

Synopsis

procedure XADsetpos(id:integer, x:integer, y:integer, w:integer, h:integer)

Arguments

id Object identifier

x x coordinate

y w coordinate

w Width

h Height

Further information

This procedure repositions the object *id*. Note: use the XADget* routines to leave parameters unchanged.

Related topics

[XADgeth](#), [XADgetw](#), [XADgetx](#), [XADgety](#)

XADsettext

Purpose

(Re)set the textual information of an object.

Synopsis

```
procedure XADsettext(id:integer, text:string)
```

Arguments

id Object identifier

text New textual information

Further information

This procedure updates the textual information of an object if applicable.

XADsetvisible

Purpose

Making an object visible/hidden.

Synopsis

procedure XADsetvisible(id: integer, visible: boolean)

Arguments

id Object identifier

visible Display option.

 true object visible

 false object hidden

Further information

Objects may be shown or hidden by calling this procedure.

XADgeteventtext

Purpose

Retrieve the message associated with an event.

Synopsis

```
function XADgeteventtext:string
```

Return value

The event message or an empty string.

Further information

If an event carries textual information (for example a tab selection), use this function to retrieve the text when handling the event.

Related topics

[XAD_EVENT_CHANGED](#), [XAD_EVENT_SELECTION](#), [XAD_EVENT_KEYDOWN](#),
[XAD_EVENT_KEYUP](#)

Chapter 7

Generic events

XAD_EVENT_KEYDOWN	Key pressed.	p.90
XAD_EVENT_KEYUP	Key released.	p.91
XAD_EVENT_MOUSE_LEFTDOWN	Left mouse button pressed.	p.91
XAD_EVENT_MOUSE_LEFTUP	Left mouse button released.	p.91
XAD_EVENT_MOUSE_MOVED	Mouse moved.	p.91
XAD_EVENT_MOUSE_RIGHTDOWN	Right mouse button pressed.	p.92
XAD_EVENT_MOUSE_RIGHTUP	Right mouse button released.	p.92

XAD_EVENT_KEYDOWN

Description Key pressed.

Notes Indicates that a key was *pressed* when the object had the focus. Call [XADgeteventtext](#) to obtain the representation of the key that was pressed. Letters and digits are returned as themselves. The following special codes can also be returned:

Up arrow	"up"
Down arrow	"down"
Left arrow	"left"
Right arrow	"right"
Tab	"tab"
Enter	"enter"
Shift	"shift"
Ctrl	"control"
Caps Lock	"capslock"

Esc	"esc"
Spacebar	" "
Page Up	"pageup"
Page Down	"pagedown"
End	"end"
Home	"home"
Insert	"ins"
Delete	"del"

XAD_EVENT_KEYUP

Description Key released.

Notes Indicates that a key was *released* when the object had the focus.
See [XAD_EVENT_KEYDOWN](#) for more information.

XAD_EVENT_MOUSE_LEFTDOWN

Description Left mouse button pressed.

Notes Indicates that the left mouse button was *pressed* above the object. The coordinates of the mouse relative to ANY object *id* can be obtained ANYTIME using `XADgetmousex` and `XADgetmousey`.

XAD_EVENT_MOUSE_LEFTUP

Description Left mouse button released.

Notes Indicates that the left mouse button was *released* above the object.

XAD_EVENT_MOUSE_MOVED

Description Mouse moved

Notes Indicates that the mouse has *moved* above the object.

XAD_EVENT_MOUSE_RIGHTDOWN

Description Right mouse button pressed.

Notes Indicates that the right mouse button was *pressed* above the object.

XAD_EVENT_MOUSE_RIGHTUP

Description Right mouse button released.

Notes Indicates that the right mouse button was *released* above the object.

Chapter8

Utility routines

XADchoosefile	Display file selection dialog.	p.96
XADhandleevents	Handling events during program execution.	p.95
XADpopupmenu	Creation of a pop-up menu.	p.97
XADseteventcallback	Set the event handler callback.	p.94

XADseteventcallback

Purpose

Set the event handler callback

Synopsis

```
procedure XADseteventcallback(handlername:string)
```

Argument

 handlername Event handler callback

Further information

This procedure registers the procedure *handlername* to act as an event handler callback. All events will be reported through this callback. The callback procedure has this signature:

```
procedure guievents(id:integer, event:integer)
```

and it gets called for every possible event. The user has the option to ignore or deal with events through the event handler callback procedure.

XADhandleevents

Purpose

Handling events during program execution.

Synopsis

procedure XADhandleevents

Further information

If a long calculation (*e.g.* optimization) is initiated by the event handler callback, the user interface will freeze. This is due to the fact that the processor intensive operations take place on the same thread as the code that draws the user interface or responds to user events (after all, we are dealing with an event). To avoid this phenomenon, call `XADhandleevents` at regular intervals (*e.g.* during Optimizer callbacks) to allow the user to interact with the user interface. Use caution, however as the following call sequence is likely to occur:

```
    guievents(1,1)
calls  minimize(objective)
calls  globallog
calls  XADhandleevents
calls  guievents(1,10)
calls  ???
```

While an optimization is running events should be dealt with quickly and with little (if well understood) or no side effects. In this situation one could use [XADseteventcallback](#) to switch to an alternative, simplified event handler callback.

Related topics

[XADseteventcallback](#)

XADchoosefile

Purpose

Display file selection dialog.

Synopsis

```
function XADchoosefile(opensave: boolean): string
```

Argument

 opensave Dialog type selection.

 true create an *Open file* dialog

 false create a *Save file* dialog

Return value

File name if selection was successful, otherwise an empty string.

Further information

This is a convenience routine for displaying the standard Windows file selection dialog. Pass `true` as an argument to create an *Open file* dialog and `false` to create a *Save file* dialog. If the file selection was successful, the returned string contains the file name. If not, the function returns an empty string.

XADpopupmenu

Purpose

Creation of a pop-up menu.

Synopsis

```
function XADpopupmenu(menuitems:set of string):string
```

Argument

 menuitems Menu items

Return value

Selected item, or empty string if no selection was made.

Example

To create a menu with the items *Action A* and *Action B* separated by a line, use the following:

```
choice:=XADpopupmenu({"ActionA", "XADseparator", "Action B"})
```

Further information

It is customary for user interfaces to display a menu of options when the user right-clicks on something. This routine achieves just that. When the user makes a choice, it is returned as a string. If the user does not select anything, the returned string will be empty.

Chapter 9

Example

XAD が Mosel とどのように連携して、対話型の数学的プログラミングモデルを作成するかを説明するため、簡単な割当ての問題を使います。

以下は、アプリケーションの画面ショットです：

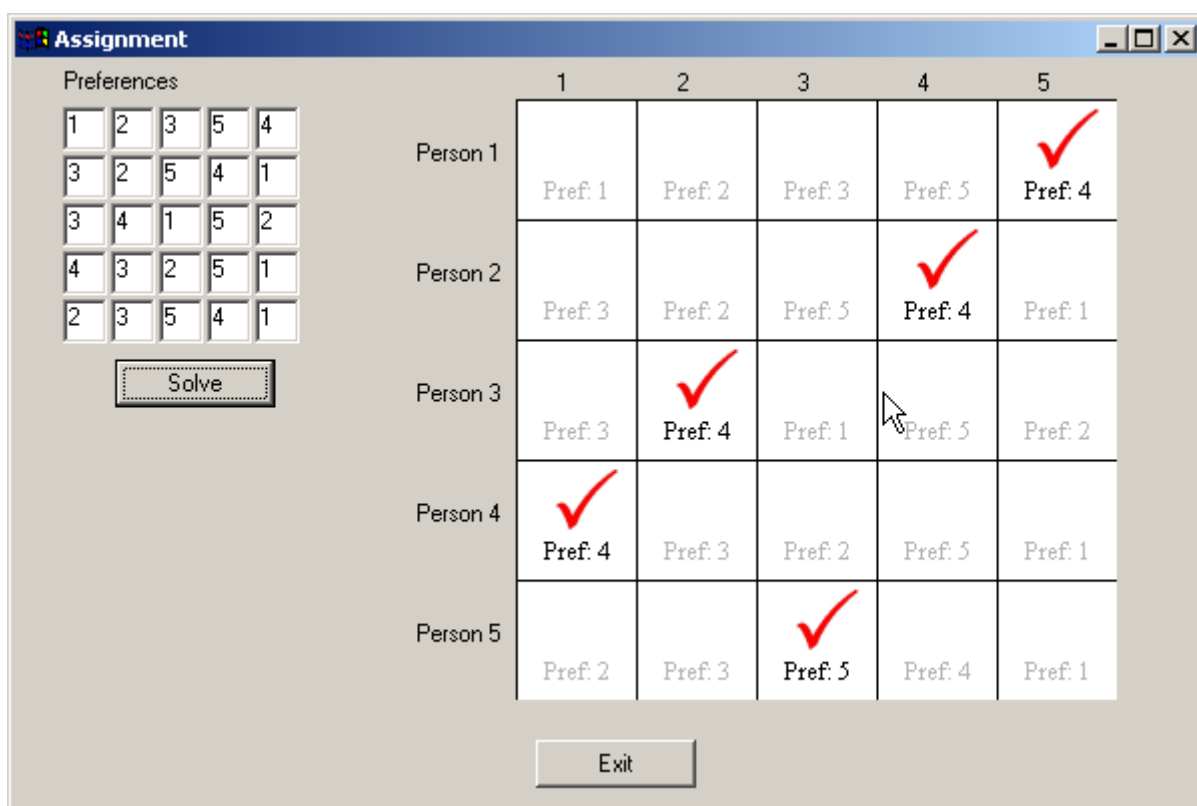


図 9.1: 対話型数学的プログラミングモデル

ユーザーは左上の角にある入力ボックスの数字を変更して優先度を変更することができます。*Solve* を押すと、結果が図表に表示されます。この手順は、何度でも繰り返すことができます。

コードを一つ一つ調べてみましょう：

```

model Assignment
  uses "mmxprs", "mmxad"

  declarations
    NP = 5                                ! Number of persons/projects
    RP = 1..NP                            ! Set (range) of persons/projects
    PREF: array(RP,RP) of integer         ! Preference values
    assign: array(RP,RP) of mpvar        ! Assignment person-project
  end-declarations

```

このアプリケーションを構築するためには、XAD(*mmxad*)だけでなく Xpress オプティマイザ (*mmxprs*)も使う必要があります。また、決定変数の配列 *assign* のような後でモデル中で使用する識別子をいくつか宣言します。

次の部分では:

```

11  PREF:= [1, 2, 3, 5, 4,
12          3, 2, 5, 4, 1,
13          3, 4, 1, 5, 2,
14          4, 3, 2, 5, 1,
15          2, 3, 5, 4, 1]
16  ! Objective function: maximize satisfaction
17  Satisfaction:= sum(m,p in RP) PREF(m,p) *assign(m,p)
18  ! One person per project
19  forall(p in RP) OnePersProj(p) := sum(m in RP) assign(m,p)=1
20  ! One project per person
21  forall(m in RP) OneProjPers(m) := sum(p in RP) assign(m,p)=1

```

優先度の例を割り当てます(これはユーザーが編集することができます)。目的関数 *Satisfaction* は、割当てがされるかどうかに依存する優先度のすべての合計を示します。2つの制約、つまり1プロジェクトにつき1人、1人につき1プロジェクトという制約によってこのモデルの数学的公式化が完了します。

ユーザーインターフェースのコードを調べてみましょう:

```

23  declarations
24      id_win=1;          id_textprefs=2;      id_buttonsolve=3;
25      id_canvas=4;      id_buttonexit=5;     id_inputs=10; id_texts=100;
26  end-declarations
27
28  X&Dcreatewindow      (id_win,100,100,600,400,          "Assignment")
29  X&Dcreatetext        (id_win,id_textprefs,24,4,65,15,   "Preferences")
30  forall(i in 1..NP,j in 1..NP) do
31      X&Dcreateinput   (id_win,id_inputs+10*i+j,24*j,24*i,22,22, ""+PREF(i,j))
32  end-do
33  X&Dcreatebutton      (id_win,id_buttonsolve,50,150,80,24, "Solve")
34  forall(i in 1..NP) do
35      X&Dcreatetext    (id_win,id_texts+(NP+i),270+(i-1)*60,5,15,15, ""+i)
36      X&Dcreatetext    (id_win,id_texts+i,200,40+(i-1)*60,45,15, "Person "+i)
37  end-do
38  X&Dcreatecanvas      (id_win,id_canvas,250,20,300,300)
39  X&Dcreatebutton      (id_win,id_buttonexit,260,340,80,24, "Exit")

```


上記のコードでは、一意の ID を種々の GUI オブジェクトに割り当てるための宣言を、新たにまとめてしています。まずウィンドウを生成し、次に `id_win` を親として XAD オブジェクトを一つずつ生成します。`id_inputs` および `id_texts` は、1 以上のアイテムに対して一意の ID を生成するために、`i` および `j` と組み合わせて使用しているという意味で、特別である点に注意してください。また、`" "+PREF(i,j)` という表現が、実際どのように各入力オブジェクトに、対応する優先度の評価を入れるのかにも注意してください。

すべての GUI オブジェクトの座標と大きさは、Xpress アプリケーションディベロッパーデザイナーツールを使って得ることができます。このツールは *What You See Is What You Get* の GUI エディタとして動く Mosel プログラム (XAD で記述されたもの) です。最終結果に納得がいくまでに、何度かレイアウトの試験が必要となるかもしれません。

XAD オブジェクトに対する ID として整数を使用すると、理解しやすく、(上記のすべての入力のような)カテゴリーの管理がしやすいオブジェクトのグループ化が容易になります。特に大きなモデル/アプリケーションでこの特徴を活かしてください。

キャンバス `id_canvas` は、今の最適化問題の結果を表示します。プロシージャ全体で、このオブジェクトを最新の情報に更新します:

```

41 procedure UpdateCanvas
42     XADcanvaserase(id_canvas,XAD_WHITE)
43     !draw the assignment table
44     forall(i in 1..NP) do
45         XADcanvasdrawline(id_canvas,(i-1)*60,0,(i-1)*60,300,XAD_BLACK)
46         XADcanvasdrawline(id_canvas,0,(i-1)*60,300,(i-1)*60,XAD_BLACK)
47         forall(j in 1..NP) do
48             if getsol(assign(i,j))<>0 then
49                 XADcanvasdrawimage(id_canvas,(j-1)*60+20,(i-1)*60+5,31,30,
50                                     "checkmark.bmp")
51                 XADcanvasdrawtext(id_canvas,(j-1)*60+30,(i-1)*60+45,
52                                   "Pref: "+PREF(i,j),XAD_BLACK)
53             else
54                 XADcanvasdrawtext(id_canvas,(j-1)*60+30,(i-1)*60+45,
55                                   "Pref: "+PREF(i,j),XADcolor(180,180,180))
56             end-if
57         end-do
58     end-do
59     XADcanvasrefresh(id_canvas)
60 end-procedure

```

一般に、キャンバスは最初に `XADcanvaserase` を使って消去してください。すべての描画が終わった後、`XADcanvasrefresh` を呼んで、キャンバスの内容を更新してください。

`UpdateCanvas` プロシージャはグリッドを描画し、次に配列 `assign` の最適化された値を基にして、各セルを更新します。割り当てられる場合には、セルにビットマップ画像ファイルのチェックマークも描画します。割り当てられない場合には、セルのテキストを、強調しないように薄く陰のかかった灰色で描画します。

では、イベントを処理するコールバックプロシージャおよびアプリケーションを開始するコード

を調べましょう。

```
62 procedure guievents(id:integer, event:integer)
63   if id=id_buttonsolve and event=XAD_EVENT_PRESSED then
64     /update preferences
65     forall(i in 1..NP,j in 1..NP) do
66       PREF(i,j):=integer(XADinputgettext(id_inputs+10*i+j))
67     end-do
68     /update objective function
69     Satisfaction:= sum(m,p in RP) PREF(m,p)*assign(m,p)
70     maximize(Satisfaction)
71     /update canvas
72     UpdateCanvas
73   elif id=id_buttonexit and event=XAD_EVENT_PRESSED then
74     XADwindowclose(id_win)
75   elif id=id_win and event=XAD_EVENT_WINDOW_OPENED then
76     UpdateCanvas
77   end-if
78 end-procedure
79
80 XADseteventcallback("guievents")
81 XADwindowopen(id_win)
82
83 end-model
```

このアプリケーションでは、ウィンドウを開いたとき、および 2 つのボタンのいずれかを押したときの 3 つのイベントが重要です。

id_buttonsolve ボタンを押すと、Mosel は優先度の配列を更新し(入力オブジェクトから直接取得します)、新しい優先度に基づいて目的関数を更新し、問題を最適化し、最後に一連の新しい割当てを表示するためにキャンバスを更新しなければなりません。

id_buttonexit ボタンを押すと、直ちにウィンドウを閉じます。

[XAD_EVENT_WINDOW_OPENED](#) イベントは、ユーザーがウィンドウを操作する機会を得る前に、ウィンドウの状態を更新するために処理をしてください。これはウィンドウのライフタイムで最初のイベントです。

Moselコードの中にあと 2 つ重要なステートメントがあります。[XADseteventcallback](#) は、必ずウィンドウを開く前に呼ばなければなりません。ウィンドウがイベントを送ることができるようにするためです。最後に[XADwindowopen](#) でウィンドウを開き、(イベントを通しての)実行の制御をウィンドウに渡します。

[XADwindowclose](#) を呼んだとき、またはマウスでウィンドウを閉じたとき、Moselコードの実行は、実際[XADwindowopen](#) に続くステートメントを継続する点に注意してください(この例では、他に実行するものがないためアプリケーションは終了します)。これは、ウィンドウがイベントハンドラーを通してMoselの実行を一時的に制御する、Moselアプリケーションの単なるコンポーネントであると常に考えるべきであることを意味します。Moselが常に管理を行い、いつでもウィンドウを終わらせることができます。

(XADアプリケーションを Xpress-IVE で開発する場合、プログラムの実行を終わらせるため

に停止機能を使用しないでください。Mosel と IVE が不正な状態になり、ユーザーオブジェクトを扱うシステムコールの間、Mosel の実行を中断することがあります。停止を使用する代わりに、すべての XAD ウィンドウは常に閉じてください。)

この他の XAD アプリケーションの例については、XAD に伴う一連の例を参照してください。