Delphi ApplicationへのWinRunnerの適用手法の検討

# <u>WinRunnerの概要</u>

オブジェクト認識の仕組み

# <u>Delphi対応の概要</u>

オブジェクトの分類 Add-inの種類と概要



Add-in適用パターンの比較



## オブジェクト認識の仕組み

Delphi ApplicationへのWinRunnerの適用手法の検討



# Delphi ApplicationへのWinRunnerの対応

# <u>WinRunnerにDelphi objectを登録</u>

### 要件

- ・各<u>オブジェクト</u>をGUIマップに登録出来る (自動記録・学習)
- •スクリプトの記録/再生が可能

- ・標準的オブジェクト
- ・カスタムオブジェクト
- ・認識不能オブジェクト

Delphi Applicationへの

Application

Delphi ApplicationへのWinRunnerの適用手法の検討

\_ 🗗 🗙

WinRunnerの提供するDelphi Add-in

Script テスト対象Application Dlphinit WinRunner 本体 構 築 (Build) Script (wrun.exe) Dlphlib Delphi コンパイラ 記録関数 TestSrvr Application (Dlph\_ext.dll) Delphi Unit File Application For WR Add in Source file 他にもdiblib.dll,dlph\_gvf.dll,dlph\_tbl.dll等





#### 

# Map all standard Delphi classes

### 

add\_dlph\_obj ("TForm","window", "class MSW\_class dlph\_name", "label","Count\_objects", DLPH\_OBJ); add\_dlph\_obj ("TRadioGroup", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TDBRadioGroup", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TGroupBox", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TDBNavigator", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TDBNavigator", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TDBImage", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TDBImage", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TBitbtn", "push\_button", "class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TButton", "push\_button", "class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TBotton", "push\_button", "class dlph\_name", "dlph\_parent","Enabled", DLPH\_OBJ); add\_dlph\_obj ("TButton", "push\_button", "class MSW\_class dlph\_name", "dlph\_parent","Selection", DLPH\_OBJ); add\_dlph\_obj ("TDBComboBox", "combobox", "class MSW\_class dlph\_name", "dlph\_parent","Selection", DLPH\_OBJ); add\_dlph\_obj ("TDBComboBox", "combobox", "class MSW\_class dlph\_name", "dlph\_parent","Selection", DLPH\_OBJ);

#### 

# Grid Objects

#### 

add\_dlph\_obj ("TStringGrid","object", "class MSW\_class dlph\_name", "dlph\_parent","", DLPH\_GRID); add\_dlph\_obj ("TDBGrid","object", "class MSW\_class dlph\_name", "dlph\_parent","", DLPH\_GRID); add\_dlph\_obj ("TDBGridInplaceEdit","object", "class MSW\_class dlph\_parent", "","Enabled", DLPH\_GRID); add\_dlph\_obj ("TPopupGrid","object", "class MSW\_class dlph\_name", "dlph\_parent","", DLPH\_GRID);

#### 

# Custmerszie Objects

#### 

add\_dlph\_obj ("TPopupListbox","object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TPopupDataList","object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TDBLookupListBox", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TDBLookupComboBox", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TPanel", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TPanel", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ); add\_dlph\_obj ("TToolbar97", "object", "class MSW\_class dlph\_name", "dlph\_parent","Enabled", DLPH\_REC\_OBJ);

## Delphi Add-in(Add-in Loaded)TypeAの有無による 自動記録スクリプトの相違

### Delphi ApplicationへのWinRunnerの適用手法の検討



## Delphi Add-inの種類による自動記録スクリプトの相違













##### TSL(Test Script Language) ####
button\_press ("SpeedButton");

Click "SpeedButton1"



#### TSL(Test Script Language) #### #Pattern\_A修正用初期化スクリプト-##Customized object set\_class\_map("F8ToolBarClass", "toolbarwindow32"); set\_record\_attr("F8ToolBarClass", "class MSW\_class", "label", "location"); set\_record\_method("F8ToolBarClass", RM\_RECORD); set\_class\_map("TF8CheckListBoxClass", "listbox"); set\_record\_attr("TF8CheckListBoxClass", "class MSW\_class", "label", "location"); set\_record\_method("TF8CheckListBoxClass", RM\_RECORD); set\_class\_map("TF8Panel", "object"); set\_record\_attr("TF8Panel", "class MSW\_class", "label", "location"); set\_record\_method("TF8Panel", RM\_RECORD); set\_class\_map("TF8Edit", "edit"); set\_record\_attr("TF8Edit", "class MSW\_class", "label", "location"); set\_record\_method("TF8Edit", RM\_RECORD); set\_class\_map("TF8ComboBox", "combobox"); set\_record\_attr("TF8ComboBox", "class MSW\_class", "label", "location"); set\_record\_method("TF8ComboBox", RM\_RECORD); set\_class\_map("TF8ListView", "syslistview32"); set\_record\_attr("TF8ListView", "class MSW\_class", "label", "location"); set\_record\_method("TListView", RM\_RECORD); ##Delphi standard object set\_class\_map("TToolBar", "toolbarwindow32"); set\_record\_attr("TToolBar", "class MSW\_class", "label", "location"); set\_record\_method("TToolBar", RM\_RECORD); set\_class\_map("TSpinEdit", "object"); set\_record\_attr("TSpinEdit", "class MSW\_class", "label", "location"); set\_record\_method("TSpinEdit", RM\_RECORD); set\_class\_map("TPanel", "object"); set\_record\_attr("TPanel", "class MSW\_class", "label", "location"); set\_record\_method("TPanel", RM\_RECORD); set\_class\_map("TPageControl", "SysTabControl32"); set\_record\_attr("TPageControl", "class MSW\_class", "label", "location"); set\_record\_method("TPageControl", RM\_RECORD); set\_class\_map("TComboBox", "combobox"); set\_record\_attr("TComboBox", "class MSW\_class", "label", "location"); set\_record\_method("TComboBox", RM\_RECORD); set\_class\_map("TListBox", "listbox"); set\_record\_attr("TListBox", "class MSW\_class", "label", "location"); set\_record\_method("TListBox", RM\_RECORD); set\_class\_map("TListView", "syslistview32"); set\_record\_attr("TListView", "class MSW\_class", "label", "location"); set\_record\_method("TListView", RM\_RECORD); set\_class\_map("TTreeView", "systreeview32"); set\_record\_attr("TTreeView", "class MSW\_class", "label", "location"); set\_record\_method("TTreeView", RM\_RECORD); add\_dlph\_obj ("TListView", "syslistview32", "class MSW\_class dlph\_name", "dlph\_parent", "Selection", DLPH\_OBJ); add\_dlph\_obj ("TToolBar", "toolbarwindow32", "class MSW\_class dlph\_name", "dlph\_parent", "Enabled", DLPH\_OBJ); #dlpj\_Panel\_Button\_Pressが記録をHookしてしまうのでここで再定義してやる必要有 add\_dlph\_obj ("TPanel", "object", "class MSW\_class dlph\_name", "dlph\_parent", "Enabled", DLPH\_OBJ); #Pattern\_C修正用初期化スクリプト------add\_dlph\_obj ("TF8CheckListBoxClass", "listbox", "class MSW\_class dlph\_name", "dlph\_parent", "Selection", DLPH\_OBJ); add\_dlph\_obj ("F8ToolBarClass", "toolbarwindow32", "class MSW\_class dlph\_name", "dlph\_parent", "Enabled", DLPH\_OBJ); add\_dlph\_obj ("TF8Panel", "object", "class MSW\_class dlph\_name", "dlph\_parent", "Enabled", DLPH\_OBJ); add\_dlph\_obj ("TF8Edit", "edit", "class MSW\_class dlph\_name", "dlph\_parent", "Compare", DLPH\_OBJ); add\_dlph\_obj ("TF8Edit", "edit", "class MSW\_class dlph\_name", "dlph\_parent", "Selection", DLPH\_OBJ); add\_dlph\_obj ("TF8ComboBox", "combobox", "class MSW\_class dlph\_name", "dlph\_parent", "Selection", DLPH\_OBJ); add\_dlph\_obj ("TListView", "syslistview32", "class MSW\_class dlph\_name", "dlph\_parent", "Selection", DLPH\_OBJ); add\_dlph\_obj ("TToolBar", "toolbarwindow32", "class MSW\_class dlph\_name", "dlph\_parent", "Enabled", DLPH\_OBJ);











表2 Add-in適用バターン別 GUIマップ及び目
----------------------------

Add-in	GUIマップ		
適用パターン	物理的記述	論理名	スクリプト
Pattern_A	・メニュー項目を除く全てのオブジェクトが objectクラスにマッピングされる(*1)。 ・HWNDやコントロールIDが物理的記述に採用された場合、 再生時認識不能。	″クラス名″ or ″labelプロパティ″(*2)	ほとんど全ての操作を ″obj_mouse_click″ or ″obj_type″ で記録。
Pattern_B	・dlph_init内で定義済みのオブジェクトは 各々最適なクラスにマッピングされる。 ・カスタムオブジェクトは全てobjectクラスにマッピングされる。 このようなオブジェクトはPattern_Aと同様認識不能となる場合有り。	同上	マッピングされたクラス固有の関数で記録。 button_press(); eidt_set(); tab_select_item(); 等
Pattern_C	同上	Delphオブジェクトの Nameプロパティ	Pattern_Bの内容に加え、 PanelやGridの操作関数が記録される。 dlph_panel_button_press(); tbl_get_cell_data(); 等

(\*1)#32770等のOSに用意されたダイアログを除く

(\*2)labelプロパティは多くのDelphiオブジェクトには存在しないが、caption等がlabelとして採用されている

### 表3 Add-in適用パターン別 テスト修正方法の特徴

Add-in 適用パターン	修正方法	修正対象オブジェクト
Pattern_A	オブジェクト毎に類似クラスにマッピングする関数を 初期化スクリプトに追加する。	ほぼ全てのオブジェクト
Pattern_B	同上	カスタムオブジェクト及び dlph_initに含まれていないDelphiオブジェクト
Pattern_C	同上	同上

Add-in適用パターン別 コスト比較

表4 Ad	!ーin適用ノ	パターン	′別 コ	スト比較
-------	---------	------	------	------

Add−in 適用パターン	テスト作成・維持コスト	テスト実行時コスト	ツール自体のコスト
Pattern_A	× (*1)	_	O(*5)
Pattern_B	O(*2)	—	× (*6)
Pattern_C	©(*3)	× (*4)	× (*6)

(\*1)ほぼ全てのオブジェクトが自動ではobjectクラスとしてしか認識されない。

- (\*2)標準Delphiオブジェクトに関しては最適なクラスにマッピングされる。
- (\*3)・GridやPanelオブジェクトに対して、個別の便利な関数が使用可能となる。 ・Nameプロパティを取得出来る。
- (\*4) ApplicationにTestUnitを組込んで、再構築する必要有り。詳細は"5.2 業務フロー"参照。
- (\*5) WinRunner本体以外のコストは発生しない。
- (\*6) 組込み用TestUnit(TestSrvr)は別売り。購入及び維持費用が別途発生。