



成都理工大学

地球物理学院

# 《地球与空间探测数据处理方法》

## 第七章 图形用户界面

---

陶 丹

Email: [adam.tao@hotmail.com](mailto:adam.tao@hotmail.com)

Address: 北翼楼（地物院）5814室



## □ 课程基础

高等数学/线性代数/概率论与数理统计

高级程序语言与程序设计(C)/Matlab 程序设计

## □ 学习目的

1. 掌握IDL可视化分析工具的基础内容，能够独立程序编写并处理地球空间数据。
2. 能够熟悉了解空间探测数据类型及数据文件的存储格式，并能够熟练利用IDL可视化工具来实现空间探测数据的读写、数据格式的转换以及相关可视化分析等。
3. 能够熟练掌握空间探测数据的坐标变化、数据的平滑处理、滤波、谱分析、相关性分析、拟合、平均值统计、中值及误差分析等方法。

## □ 考核方式

**平时成绩 (40%) + 考试成绩 (60%)**

其中：

平时成绩 (40%)：考勤 (10%) + 大作业/实验报告 (30%)

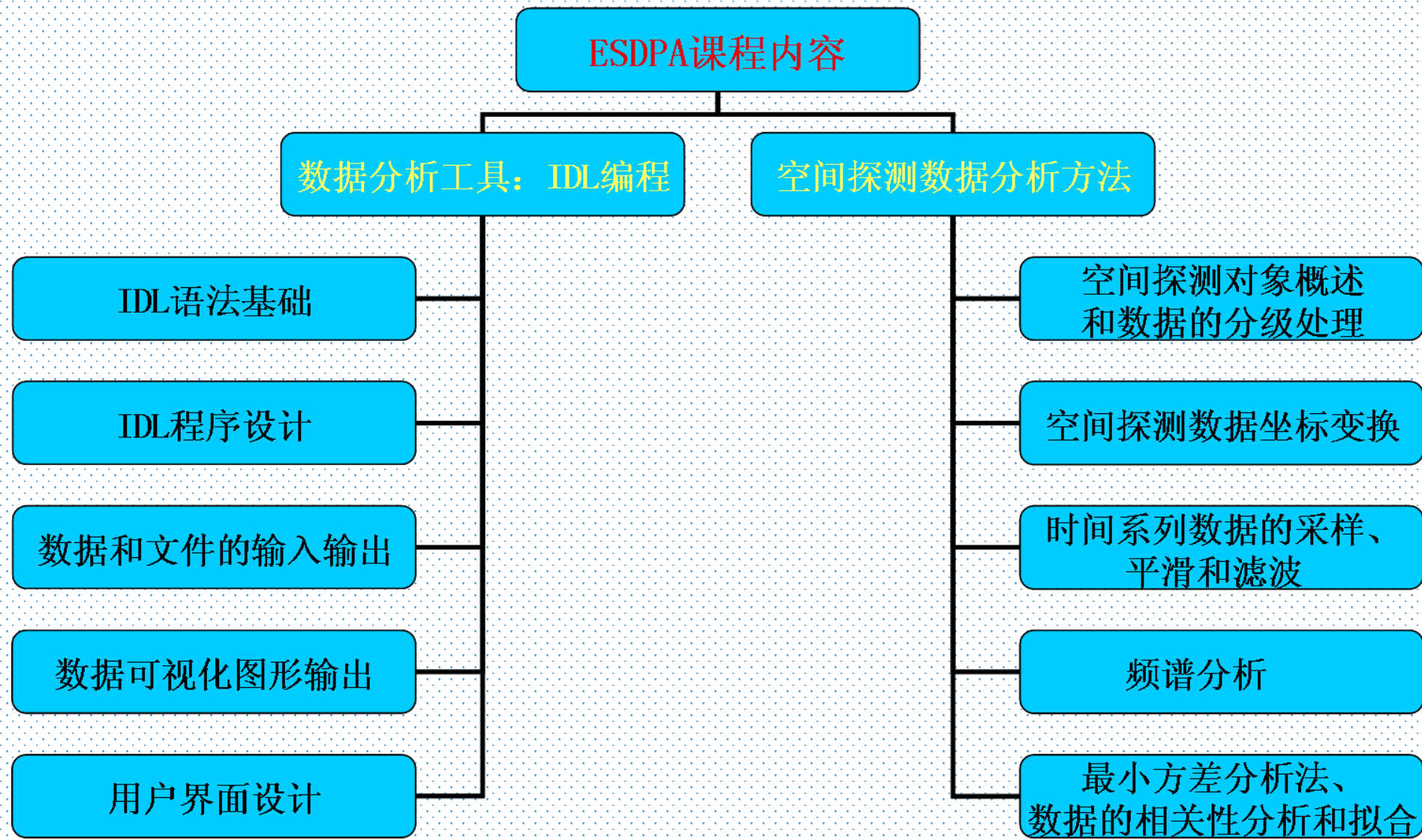
考试成绩 (60%)：期末考试 (60%，闭卷)

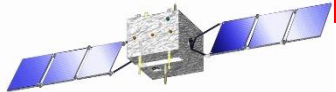


- 《IDL可视化工具入门与提高》，闫殿武编著，机械工业出版社，2003.
- 《IDL可视化分析与应用》，韩培友编著，西北工业大学出版社，2006.
- 《IDL程序设计—数据可视化与ENVI二次开发》，董彦卿编著，高等教育出版社，2012.



## 地球与空间探测数据处理方法课程架构





## 七、图形用户界面：GUI的用途和组成

图形用户界面**GUI** (**G**raphical **U**ser **I**nterface): 是用户与应用程序之间进行交互控制和相互传递数据和信息的图形界面。

### ■ GUI用途

把复杂的程序控制转化为简单的GUI，用户可以方便、灵活、快捷地对复杂的程序进行控制。用户可以不熟悉IDL源程序的编程方法，只需要知道GUI软件界面的功能和使用方法即可。

### ■ GUI组成

由菜单系统、工具栏和大量的组件构成。组件是指用来表示用户与程序输入输出交互作用的图形元素。按钮、滑动条和文本框都是组件。组件工具箱是一套IDL API命令的集合，它主要用于开发图形界面程序。IDL支持9种基本的组件类型。

基本的组件类型列表

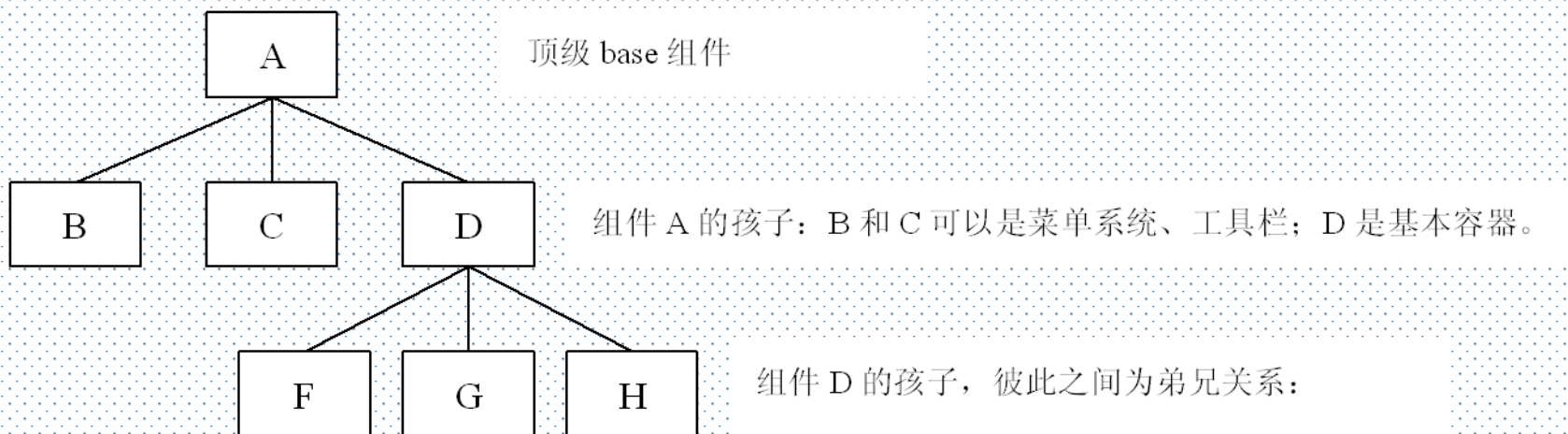
组件	描述	创建函数
Base	装放其它组件的容器	Widget_base()
Button	可单击按钮	Widget_button()
Draw	绘图窗口	Widget_draw()
Droplist	单击后打开一个可选择列表的按钮	Widget_droplist()
Label	不可编辑的文本标签	Widget_label()
List	包含可选择的列表，通常有滚动条	Widget_list()
Slider	水平或垂直滚动条	Widget_slider()
Table	可编辑表格	Widget_table()
Text	可编辑文本区域	Widget_text()



## 七、图形用户界面：GUI的组成结构

### ■ GUI组件之间的关系

- GUI中首先创建的组件必须是顶级基本容器A (**BASE**)，顶级基本容器是最上层的基本容器，它包含后面创建的所有组件，每一个GUI只能创建一个顶级基本容器A。
- 基本容器D是包含在顶级基本容器A中的孩子组件，它也可以包含其他的任意组件,譬如触发按钮、滑动条、绘图区、文本编辑框、文本标签、列表框、下拉列表框、树结构组件、页面组件或其他基本容器等。



组件的层次结构图





## 七、图形用户界面：GUI的设计步骤

由于GUI是用户和应用程序之间进行交互控制和相互传递数据和信息的重要桥梁，所以GUI设计的好坏将直接应用程序的功能实现。因此在设计GUI时，首先要认真分析应用程序的模块结构和用户需求，然后再根据用户需求和应用程序结构设计出合理的GUI。

### ■ GUI的设计步骤：

- 1 通过对用户需求的分析，给出应用程序的功能模块及其组成结构；
- 2 编写好应用程序中相关过程或函数的核心部分，并分清每个过程或函数对应的GUI组件或事件；
- 3 根据GUI组件之间的关系，给出结构和布局合理的GUI设计方案；
- 4 使用基于过程的GUI设计方法或基于GUIBUILDER的设计方法设计并实现GUI；
- 5 激活GUI；
- 6 对GUI进行循环控制；
- 7 利用顶级BASE的UVALUE建立用于传递数据的结构变量；
- 8 对GUI的组件性质、事件进行设置和控制处理；
- 9 设计并实现GUI中各个组件之间的相互控制；
- 10 设计事件的具体处理方法，并设计与事件相对应的过程或函数；
- 11 设计GUI的退出事件，用于关闭GUI。



## 七、图形用户界面：创建组件

- 创建和更改组件时，有一些共同的属性可以设置。常用的组件关键字有：

关键字 (keywords)	功能
value	为该组件设置值。顶层base组件没有值，因此不支持该关键字。
uvalue	为该组件设置用户值
Group_leader=widget_id	该组领导组件的标识符。创建模态组件是必须使用。
Frame=width	设置组件边框的宽度
Sensitive=0 或 1 (/sensitive)	取0时表示该组件不敏感（即不可操作），取1时表示敏感（可操作）
Map=0 或 1 (/map)	取0时表示该组件不可见，取1时表示可见

其中：

- 组件的值VALUE定义为组件的可视设置。例如一个标签组件的值是以标签显示的文本，而一个列表组件的值是列表项的一个字符串数组。组件的值可以在组件创建时定义，也可以在组件的生命周期内修改。顶层组件是唯一没有值的组件，因此不支持VALUE关键字。
- 组件的用户值UVALUE为GUI的开发者保存，可以用来存储任何IDL变量和数据的类型。组件的用户值经常设置为一个字符串变量，该变量为组件包含一个特定的名称。组件的用户值也可以存储一条GUI应用程序状态的信息。





### ■ 基本容器组件的创建

格式：**BASE\_ID=WIDGET\_BASE**([PARENT\_ID] [, KEYWORDS])

功能：创建一个基本容器BASE组件，其返回值为该组件的身份标识ID。顶层BASE组件没有值，因此不支持VALUE关键字。

基本参数和关键字的用法：

□ **PARENT\_ID**: 给出创建BASE组件的上一级组件的ID。创建顶级BASE时，省略该参数。

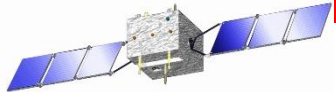
□ 顶层组件的创建：

**BASE\_MAIN=WINDGET\_BASE**([KEYWORDS])

顶层BASE组件容纳了程序中所有其它组件。事实上，它就是出现在屏幕上的、包含了程序组件界面的活动窗口。尽管顶层BASE组件在创建函数中不需要PARENT\_ID参数，但是它可能需要通过关键字来设置组件的其他属性。

□ **KEYWORDS**为一些可选关键字。除了组件常用的关键字外，其他的BASE关键字及其用法为：

Widget_base关键字	功能
Title=string	为一个 <b>顶层base</b> 组件设置标题。
Mbar=variable	为一个 <b>顶层base</b> 创建菜单系统的id，并把它存放在变量variable中。
Tlb_frame_attr=1   2   4   8   16 以上各个数值可以通过加法方式或以数值和的方式同时使用。	为一个 <b>顶层base</b> 设置边框的属性。1:不能改变base窗口的大小，或最大/最小化；2:在base组件中，禁止显示系统菜单；4:在base组件中，禁止显示标题栏；8: 不能关闭base组件；16:不能移动base组件。
/column 或column=value	设置该组件的下一级组件在该组件中按列排列，也可以通过column=value来设置列数value。
/row 或 row=value	设置该组件的下一级组件在该组件中按行排列，也可以通过row=value来设置行数value。
Xoffset=value 和 Yoffset=value	如果用在顶层base组件中，则设置离开屏幕左上角的水平和垂直距离(value的单位为像素)。如果用在子base中，则表示设置在其上一级组件中的始点的X方向和Y方向的位置。
Xpad=value 和 Ypad=value	设置该组件的下一级组件在该组件中的X向和Y向的边界位置。在使用/exclusive 或 /nonexclusive时该关键字无效。
Xsize=value 和 Ysize=value	给出组件的宽度和高度，value的单位为像素。
X_scroll_size=value 和 Y_scroll_size=value	设置组件在显示设备上的实际显示的宽度和高度。如果value(像素)值小于组件本身的宽度和高度，则为该组件添加滚动条。
/tlb_size_events	为base组件创建调整大小的事件
/align_bottom, /align_top, /align_left, /align_right 或 /align_center	设置该组件在上一级组件中的对齐方式分别为下对齐或上对齐（用于带/row的组件）、左对齐或右对齐（用于带/column的组件）、或居中。
/base_algn_bottom, /base_algn_top, /base_algn_left, /base_algn_right 或 /base_algn_center	设置该组件在下一级组件中的对齐方式分别为下对齐或上对齐（用于带/row的组件）、左对齐或右对齐（用于带/column的组件）、或居中。
/context_menu 或 /context_events	设置在该组件上右击鼠标时，产生一个弹出快捷菜单或弹出快捷菜单的事件。
Event_func=string 或 Event_pro =string	String给出在对该组件操作而产生事件时调用的函数或过程。
/exclusive 或 /nonexclusive	设置该组件中只能包含单选按钮或复选按钮组件
/floating	设置该组件为可移动的组件
Func_get_value=string	设置在使用带关键字get_value的widget_control命名对该组件的操作时所调用的函数
Pro_set_value	设置在使用带关键字set_value的widget_control命令对该组件操作时所调用的过程
/no_copy	在调用该组件时，不对该组件进行备份。节约内存。
/scroll	为组件添加水平和垂直滚动条
/toolbar	设置组件可以使用位图暗流创建工具栏
/tracking_events	设置组件的跟踪事件。即当鼠标进入或者退出组件所在区域时产生一个跟踪事件。
/units=0   1   2	设置组件的单位。0:像素（默认）；1：英寸；2：厘米
Uname=string	为组件命名一个用户名，该用户名用于后续对该组件的操作。



## 七、图形用户界面：按钮组件WIDGET\_BUTTON

- 按钮组件的创建格式：**BUTTON\_ID=WIDGET\_BUTTON** ([PARENT\_ID] [, KEYWORDS])
- 功能：创建4种形式的按钮组件，即带文本标签的按钮（默认情况下）、带有位图标签的长方形按钮、折叠式按钮、特定或非特定频道按钮，其返回值为该组件的身份标识ID。
- 按钮组件的常用关键字为：

Widget_button关键字	功能
value	设置按钮的值
Xsize=value 或 Ysize=value	设置按钮的宽度和高度（默认单位为像素）
Font=string	设置按钮上文本标签的字体
/bitmap	创建一个位图按钮。
/menu	创建一个折叠式菜单按钮
/separator	在折叠式按钮入口上方添加一隔离线
/no_release	只为特定或非特定频道按钮创建“按下”事件
/checked_menu	设置菜单系统中的菜单为开关菜单，即打勾生效，取消打勾无效
/dynamic_resize	设置按钮的大小随着文本标签的内容自动调整
/pushbutton_events	设置当鼠标单击按钮时产生一个单击事件
Tooltip=string	给出当鼠标的指针移动到该组件上时所弹出的提示信息。该关键字对图片按钮的使用比较方便。

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

**EXAMPLE-7.1**



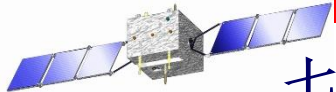
## 七、图形用户界面：绘图组件WIDGET\_DRAW

- 绘图组件的创建格式：**DRAW\_ID=WIDGET\_DRAW** ([PARENT\_ID] [, KEYWORDS])
- 功能：为用户提供一个数据可视化的场所，在绘图区可以绘制各种图形、显示各种图像和输出各种文字等。其返回值为该组件的身份标识ID。绘图组件等价一个绘图窗口。
- 绘图组件的常用关键字为：

Widget_draw关键字	功能
Xsize=value 或 Ysize=value	设置绘图组件的宽度和高度（默认单位为像素）
X_scroll_size 和 Y_scroll_size	设置绘图组件在屏幕上用于滚动显示的实际宽度和高度，而不是组件本身的大小
/button_events	在绘图组件内按下或释放鼠标左右按钮时，产生一个事件
/motion_events	当鼠标指针在绘图区内移动时，用于产生一个事件
Renderer=0   1	设置图形生成的方式。0:使用OpenGL；1：使用IDL系统
/app_scroll	创建一个带有滚动条的绘图区，通过滚动条浏览整个内容

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

### EXAMPLE-7.2



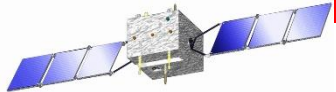
## 七、图形用户界面：下拉列表组件WIDGET\_DROPLIST

- 下拉列表组件的创建格式：**DROPLIST\_ID=WIDGET\_DROPLIST** ([PARENT\_ID] [, KEYWORDS])
- 功能：创建一个下拉列表组件。下拉列表组件允许用户从一系列之中选择某个值。下拉列表组件的值是一个字符串矩阵，包含可选列表。
- 下拉列表组件的常用关键字为：

Widget_droplist关键字	功能
value	设置下拉列表的值，该值一般为一个字符型数组
title	设置下拉列表的标题
/dynamic_resize	创建一个自动调整大小的下拉列表

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

### EXAMPLE-7.3



## 七、图形用户界面：列表组件WIDGET\_LIST

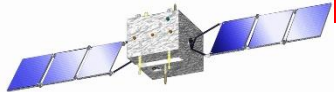
- 列表组件的创建格式：**LIST\_ID=WIDGET\_LIST** ( [PARENT\_ID] [, KEYWORDS]
- 功能：创建一个列表组件。列表组件为一个列表，用户可以从选择一个或多个选项。列表组件的值是包含可选项列表的字符串数组。
- 列表组件的常用关键字为：

Widget_list关键字	功能
value	设置列表的值，该值一般为一个字符型数组
title	设置下拉列表的标题
Xsize=value 和 Ysize=value	设置列表的宽度（字符数）和高度（项数）
/multiple	允许选择多项

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

**EXAMPLE-7.4**





## 七、图形用户界面：标签组件WIDGET\_LABEL

- 标签组件的创建格式：**LABEL\_ID=WIDGET\_LABEL** ( [PARENT\_ID] [, KEYWORDS]
- 功能：标签组件主要用于GUI中的文字说明或者状态信息的输出。标签组件在一个父母BASE组件中为不可编辑的文本字符。
- 标签组件的常用关键字为：

Widget_label关键字	功能
value	设置标签值
/align_left, /align_center, /align_right	标签左、中心或右对齐
/dynamic_resize	创建一个自动调整大小的标签
/sunken_frame	用于创建一个具有三维效果的凹陷的立体标签

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

### EXAMPLE-7.5



## 七、图形用户界面：滑动条组件WIDGET\_SLIDER

- 滑动条组件的创建格式：**SLIDER\_ID=WIDGET\_SLIDER** ([PARENT\_ID] [, KEYWORDS])
- 功能：滑动条组件主要用于交互式地获取或者设置指定范围内的一个整数值。滑动条事件发生后，用户可以读取滑动条的当前值，还可以设置滑动条的值、最小值和最大值。滑动条组件的值为当前滑动条的值。
- 滑动条组件的常用关键字为：

Widget_slider关键字	功能
value	设置滑动条的值
title	设置滑动条的标题
Maximum=value 和 Minimum=value	设置滑动条的最大和最小值
/vertical	创建一个垂直滑动条组件
/suppress_value	用于禁止显示滑动条组件的滑动块的值（压缩显示）
Scroll=units	设置当单击滑动条两端的箭头时，滑动块滑动的单位长度
/drag	当拖动滑动块时，用于产生连续的事件。正常情况下，只能产生位置事件，即当用户拖动滑动块到一个指定位置，并且释放鼠标后，才产生一个滑动条事件

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

### EXAMPLE-7.6



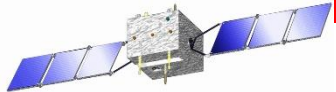
## 七、图形用户界面：文本编辑框组件WIDGET\_TEXT

- 文本编辑框组件的创建格式：**TEXT\_ID=WIDGET\_TEXT** ( [PARENT\_ID] [, KEYWORDS]
- 功能：用于交互获取或编辑文本内容。  
当文本框内容发生改变后，会产生一个事件。
- 文本编辑框组件的常用关键字为：

Widget_text关键字	功能
value	设置文本值
/editable	设置文本框处于可编辑的状态
/scroll	创建一个滚动文本框组件
/no_newline	用于禁止在编辑多行文本内容时在原内容后添加新行
/wrap	设置文本框处于自动换行状态
/all_events	当用户改变文本框内容是，产生一个事件，如果使用了/editable可省略该关键字

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

**EXAMPLE-7.7**



## 七、图形用户界面：表格组件WIDGET\_TABLE

- 表格组件的创建格式：**TABLE\_ID=WIDGET\_TABLE( [PARENT\_ID] [, KEYWORDS]**
- **功能：**用于交互获取或编辑表格内容。当表格内容发生改变后，会产生一个事件。表格组件以表格的形式显示一个两维数组或一个结构体矢量，其中的每一个单元格包含一个值。表格组件的值可以是任何数字类型或字符串类型的两维数组或一个结构体矢量。
- 表格组件的常用关键字为：

Widget_table关键字	功能
value	设置表格的值
Alignment=0   1   2	设置表格中数据的对齐方式为：左对齐（0），居中（1）或右对齐（2）
Column_labels=string_array, row_labels=string_array	设置表格的列标签和行标签
Column_widths=array	设置表格列宽。若array为一个值，则所有列宽度相等。
Format=value	设置表中的数据格式。用法和print中的format相同
/editable	设置表格处于可编辑的状态
/no_headers	不显示表头区。但列标签和行标签仍然显示
/disjoint_selection	设置表格的表元支持不连续选择。默认值为不支持
/resizeable_columns	设置表格的列宽可以调整

其他关键字和WIDGET\_BASE的非特定关键字的用法相同。

### EXAMPLE-7.8



## 七、图形用户界面：GUI的激活和循环控制

### ■ GUI激活

用户创建的GUI是一个有顶层BASE(根节点)和它的下级节点组成的树状结构，GUI创建之后必须激活才能显示出来。

□ 激活格式：**WIDGET\_CONTROL, TOPBASE\_ID, /REALIZE**

□ 功能：激活并显示 TOPBASE\_ID指定的顶级BASE和它的下级节点组成的GUI。

### ■ GUI的循环控制

由于用户是通过GUI和程序之间交互控制的，所以在程序运行过程中需要多次调用GUI来实现程序的交互控制和数据传递。因此需要对GUI进行循环控制。

□ 控制格式：**XMANAGER, NAME, TOPBASE\_ID** [,CLEANUP=STRING] [,/NO\_BLOCK]  
[,EVENT\_HANDLER='PROCEDURE\_NAME' [,GROUP\_LEADER=WIDGET\_ID]

□ 功能：循环控制程序NAME中由ID指定的顶级BASE和它的下级节点组成的GUI。

- **NAME**:循环控制的过程名称;
- **TOPBASE\_ID**:循环控制的GUI的顶级BASE ID;
- **CLEANUP=STRING**: 用于给出在删除GUI时需要调用的过程名称;
- **/NO\_BLOCK**:设置在程序运行过程中，是否禁用IDL系统命令编辑行。默认为禁用。
- **EVENT\_HANDLER='PROCEDURE\_NAME'**: 用于给出在循环控制GUI中有事件发生时，需要调用的过程名称。若省略该关键字，则程序会自动调用由**NAME**、**下划线**和**EVENT**组成的默认过程。



## 七、图形用户界面：GUI的事件处理

### ■ 事件和事件结构

用户在对GUI进行循环控制的过程中，需要对GUI中的组件进行多种操作（例如：单击出发按钮），而每一种操作都会产生一个事件，用户则根据产生的事件来调用与之相对应的过程，从而实现相应的处理。等到程序运行结束后，再返回GUI，让用户进行下一操作，直到退出GUI。这就是运用程序运行的全过程。

□ **事件**：是指在GUI中对组件进行的具体操作。例如单击鼠标、移动鼠标等。由于不同组件的不同操作会产生不同的事件，所以事件本身包含了一系列的具体信息，用来记录和区分所发生的不同事件。因此用户在对事件进行处理之前，必须**分析清楚事件的结构信息，然后再对其进行相应的操作**。事件就是用户和过程（或函数）之间的桥梁。用户需要使用事件来实现过程的调用。

□ **事件结构**：事件是一个记录事件信息的结构变量，它有固定的结构。每一个事件结构有一个事件名WIDGET和3个基本成员组成。事件结构的格式如下：

EVENT\_STRUCTURE={WIDGET, ID:0L, TOP:0L, HANDLER:0L,...}

- **WIDGET**:说明事件的名称;
- **ID**:给出产生事件的组件的ID;
- **TOP**:产生事件的顶级BASE的ID;
- **HANDLER**: 给出产生事件后所调用的函数或者过程的句柄，即**函数或过程的标识ID**。

### ■ 事件处理

□ 创建组件时，使用产生事件的关键字指定产生事件后调用的过程或函数。

[, **EVENT\_FUNC**=STRING] [, **EVENT\_PRO**=STRING] [, **FUNC\_GET\_VALUE**=STRING] [, **PRO\_SET\_VALUE**=STRING] ...

例如：BUTTON\_ID=WIDGET\_BUTTON(MYBASE,EVENT\_PRO='MYPROCEDURE')

□ 在使用循环控制命令XMANAGER中，**直接指定**事件产生后所调用的过程或者函数。

□ 事件产生后，**根据事件结构变量的值**，调用相应的过程或者函数。





## 七、图形用户界面：用户值传递数据的方法

在设计程序时，为了实现各个过程之间的数据传输，通常把需要传递的数据创建一个结构变量，并利用WIDGET\_CONTROL语句把结构变量设置为顶级BASE的用户值（UVALUE）；然后在其他过程或者函数中通过获取顶级的用户值（UVALUE）来实现数据的传递。

- 利用顶级BASE用户值传递数据的三种方法为：直接法、指针法和关键字法。
- **直接法**：是指在主程序中使用关键字SET\_UVALUE的WIDGET\_CONTROL命令，把变量或者结果变量的值直接放入顶层BASE的用户值UVALUE中，然后在子程序中使用关键字GET\_UVALUE的WIDGET\_CONTROL命令，把顶层BASE的用户值UVALUE再读到指定的变量或者结构变量中来进行数据处理的数据传递方法。
- **指针法（即地址法）**：是指在主程序中使用关键字SET\_UVALUE的WIDGET\_CONTROL命令把变量或者结构变量的地址放入顶层BASE的用户值UVALUE中，然后在子程序中使用关键字GET\_UVALUE的WIDGET\_CONTROL命令，把顶层BASE的用户值UVALUE再读到指定的变量或者结构变量中，然后使用指针方法进行数据处理的数据传递方法。
- **关键字法**：是指在直接法和指针法中使用关键字/NO\_COPY（调用组件时不备份）的WIDGET\_CONTROL命令来实现主程序和子程序以及子程序和子程序之间的数据传递方法。用法如下：  
WIDGET\_CONTROL,TOPBASE,SET\_UVALUE=STATE,/NO\_COPY  
WIDGET\_CONTROL,EVENT.TOP,GET\_UVALUE=STATE,/NO\_COPY  
WIDGET\_CONTROL,EVENT.TOP,SET\_UVALUE=STATE,/NO\_COPY

### EXAMPLE-7.9



## 七、图形用户界面：其他常用的控制

□ 结构名称和成员名称的获取：

格式：TAG\_NAMES(STRUCTURE\_VARIABLE [,STRUCTURE\_NAME]

□ 组件信息的获取：

格式：INFO=WIDGET\_INFO (WIDGET\_ID [,KEYWORDS])

□ 组件的删除：

格式：WDIGET\_CONTROL,WIDGET\_ID, /DESTROY

□ 信息输出窗口：

格式：RESULT=DIALOG\_MESSAGE(MESSAGE\_TEXT [, TITLE=STRING] [,  
/DEFAULT\_CANCEL |, /DEFAULT\_NO] [, DIALOG\_PARENT=WIDGET\_ID] [,  
/ERROR |, /INFORMATION |, /QUESTION] [, /CANCEL])

功能：以模态对话框窗口的形式输出信息文本MESSAGE\_TEXT的值，其返回值为用户在模态窗口中的选择按钮的标签，即‘YES’，‘NO’，‘OK’或‘CANCEL’。



## 七、图形用户界面：其他常用的控制

### PRO Table\_widget\_example

*data=dist(7)*

*help=['Select data from the table below using the mouse.']*

*base=widget\_base(/column,title='Table\_widget\_example')*

*subbase1=widget\_base(base,/row)*

*draw=widget\_draw(subbase1,xsize=250,ysize=250)*

*subbase2=widget\_base(subbase1,/column)*

*text=widget\_text(subbase2,xs=50,ys=8,value=help,/scroll)*

*b\_value=widget\_button(subbase2,value='Show Selection Data',/dynamic\_resize)*

*b\_select=widget\_button(subbase2,value='Show Selection Cells',/dynamic\_resize)*

*b\_change=widget\_button(subbase2,value='Change to disjoint Selection mode',/dynamic\_resize)*

*b\_quit=widget\_button(subbase2,value='Quit',/dynamic\_resize)*

*table=widget\_table(base,value=data,/all\_events)*

*widget\_control,base,/realize*

*widget\_control,draw,get\_value=draw*

*;创建需要传递数据的结构变量*

*stash={draw:draw,table:table,text:text,b\_value:b\_value,b\_select:b\_select,b\_change:b\_change,b\_quit:b\_quit}*

*;利用set\_uvalue将结构变量放入顶层base的用户值中*

*widget\_control,base,set\_uvalue=stash*

*;循环控制*

*Xmanager,'Table\_widget\_example',base*

END



## 七、图形用户界面：其他常用的控制

### *PRO Table\_widget\_example\_event, ev*

*;利用get\_uvalue获取顶层base的用户值，读到stash中*

*widget\_control, ev.top, get\_uvalue=stash*

*disjoint=widget\_info(stash.table,/table\_disjoint\_selection)*

*selection=widget\_info(stash.table,/table\_select) ;get the selection of the table*

*if (selection[0] ne -1) then hasselection=1 else hasselection=0*

*if (hasselection) then widget\_control,stash.table, get\_value=value,/use\_table\_select*

*;get the data selected from the table*

*if((ev.id eq stash.table) and hasselection) then begin*

*wset,stash.draw*

*device,decomposed=0*

*loadct,39*

*plot,value,psym=-4,color=255,xrange=[0.,7.],yrange=[0.,7.],position=[0.1,0.1,0.9,0.9]*

*endif*

*if ((ev.id eq stash.b\_value) and hasselection) then begin*

*if (disjoint eq 0) then begin*

*widget\_control,stash.text,set\_value=string(value,/print)*

*endif else begin*

*widget\_control,stash.text,set\_value=string(value)*

*endelse*

*endif*



## 七、图形用户界面：其他常用的控制

```
if ((ev.id eq stash.b_select) and hasselection) then begin
  if (disjoint eq 0) then begin
    list0='Standard Selection'
    list1='Left:'+string(selection[0])
    list2='Top:'+string(selection[1])
    list3='Right:'+string(selection[2])
    list4='Bottom:'+string(selection[3])
    list=[list0,list1,list2,list3,list4]
  endif else begin
    n=n_elements(selection)
    list=strarr(n/2+1) & list[0]='Disjoint Selection'

    for j=0,n-1,2 do begin
      list[j/2+1]='Column: '+string(selection[j])+'Row: '+string(selection[j+1])
    endfor
  endelse

  widget_control,stash.text,set_value=list
endif
```



## 七、图形用户界面：其他常用的控制

```
if (ev.id eq stash.b_change) then begin
  if (disjoint eq 0) then begin
    widget_control,stash.table, table_disjoint_selection=1
    widget_control,stash.b_change, set_value='Change to standard selection mode'
  endif else begin
    widget_control,stash.table, table_disjoint_selection=0
    widget_control,stash.b_change, set_value='Change to disjoint selection mode'
  endelse
endif

if (ev.id eq stash.b_quit) then widget_control,ev.top,/destroy ;Exit

END
```





成都理工大学

CHENGDU UNIVERSITY OF TECHNOLOGY

演示完毕

Thank You

