

XT - 新增功能 #32

增加按照ribbon配置文件生成所有Command文件

2024-03-12 14:37 - 王 绪伦

状态:	已解决	开始日期:	2024-03-12
优先级:	普通	计划完成日期:	
指派给:	胡 鹤翔	% 完成:	100%
类别:		预期时间:	0.00 小时
目标版本:		耗时:	0.00 小时
描述			
<p>1. 启动功能对话框，用户浏览选中ribbon文件，指定输出目录，指定获取文件标识的函数名（比如输入框直接输入“ Document_Demo() ”）；</p> <p>2. 根据ribbon文件中所有cid="<dir>/<name>.xml"形式的action，按照Command模板在指定目录下生成对应的Command文件，Command的文件标识按Document_Demo()获取，命令的标识符为<name>，命令类为Cmd<name>；</p> <p>3. initWindow（）函数应按照cid指定的路径给出对话框配置文件的路径；</p> <pre>void Cmd<name>::initWindow() { m_pDlg->layoutBegin(); m_pDlg->loadFromXml(GetResourceFileDirPath("uisetting", "<dir>/<name>.xml")); m_pDlg->layoutEnd(); }</pre> <p>4. CommandType getType() 统一返回Command_Dialog；</p> <p>5. 命令的参数统一按照CmdParam作为成员变量给出：</p> <pre>CmdParam m_Param;</pre> <p>6. 其他必须包含的函数包括：</p> <pre>void onDataExchange(const char * diagXml); bool checkContext(); void onControlEvent(const char * ctrlName, TypeID event = Event_Any); bool checkContextParam(const TreeNode * targetNode, const TreeNode * fromNode = 0); void setContextParam(const TreeNode * targetNode, const TreeNode * fromNode = 0); ErrorCode execute(); const char * getName() const; void setParam(const CmdParam *param); CmdParam * getParam() { return &m_oParam; } void exchangeScript(ParamList * param); CommandType getType() { return Command_Dialog; } void updateFrame(); void setEditItem(const XItem * pi);</pre>			

历史记录

#1 - 2024-03-12 14:38 - 王 绪伦

- 描述 已更新。

#2 - 2024-03-12 14:56 - 王 绪伦

该功能添加到src\tools\devkit下面；

#3 - 2024-03-13 11:37 - 胡 鹤翔

- 状态从 新建 变更为 进行中

#4 - 2024-03-14 13:48 - 胡 鹤翔

- 状态从 进行中 变更为 已解决

- % 完成 从 0 变更为 100

#5 - 2024-07-16 16:56 - 王 绪伦

- 文件cmd.png 已添加

- 文件cmd2.png 已添加

- 状态从已解决 变更为 反馈

- % 完成 从 100 变更为 90

1. 生成的xml文件中如果ribbon有指定text，那么title按照text的值给定；

2.

添加强制生成选项，默认不强制生成，即在创建xml文件或者cmd文件之前检查文件是否存在，如已存在则不生成；如果用户指定强制生成，则不论文件是否存在都重新生成文件；

#6 - 2024-07-18 17:22 - 胡鹤翔

- 状态从反馈 变更为 已解决

- % 完成 从 90 变更为 100

文件

demo_ribbon.xml	386 Bytes	2024-03-12	王绪伦
cmd.png	113 KB	2024-07-16	王绪伦
cmd2.png	29.7 KB	2024-07-16	王绪伦