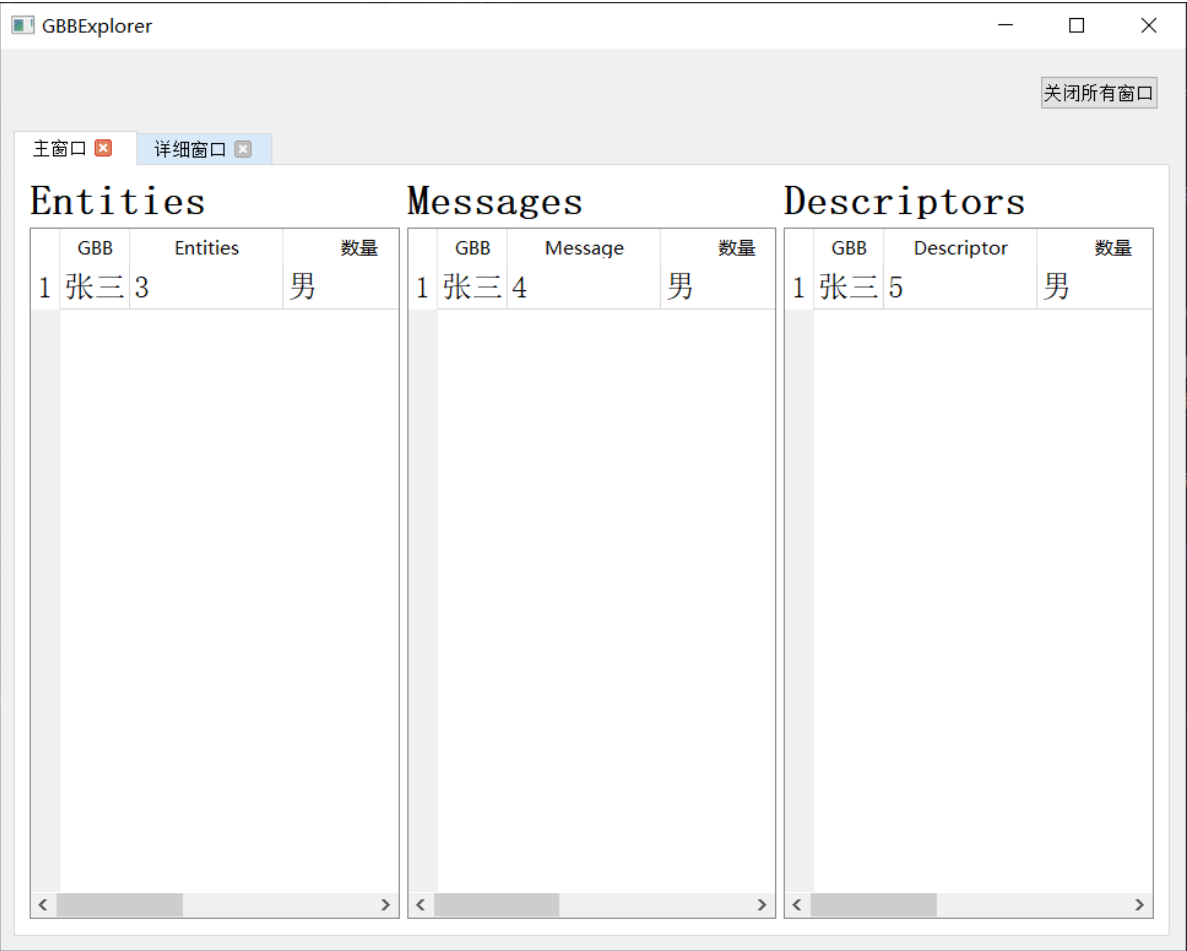


GBBExplorer记录阅读文档

Widget类，主要是绘制界面的类，界面展示如下：





在Widget.h中定义的函数和槽函数如下，实现在Widget.cpp里，其中，gbb数据未加载，这里仅用的一些预设值进行填充

具体的一些声明和注释如下

```
1 private:
2     void initForm(); //初始化主窗口
3
4
5 private slots:
6     void on_tableviewdoubleClicked(const QModelIndex &index); //双击主页
7     void on_removeTabbtn(int index); //删除标签
8     void on_pushButton_8_clicked(); //详情页全选子项目按钮的实现
9     void on_pushButton_7_clicked(); //清除按钮功能实现
10    void on_treewidget_2_clicked(QTreeWidgetItem *item); //treewidget_2选中/不
11    void on_treewidget_clicked(QTreeWidgetItem *item); //treewidget选中进行全部
    选中触发事件，模拟GBBexplorer中选择与取消
    的行显示
```

StaticData类，主要是初始化静态数据

主要参照实现了CHSim-TKE_GBBExplorer\Infra\GBBExplorer\GBBExplorer文件夹中staticData.cs和entity.cs等定义的结构，从SerializedBuffer中获取静态数据

```

1 void InitStructures(); // Create the Structures Static List
2 void InitDescriptors(); // Create the Descriptors Static List
3 void InitEntities(); // Create the Entities Static List
4 void InitMessages(); // Create the Messages Static List
5 int SetStringFromPtr(char* CurrentIntPtr, std::string &StringName);

```

main函数中

```

1 //用于日志记录和链接初始化GBB平台
2 LoggerUtil::Init();
3 if (theConfigManager.Load("UI"))
4 {
5     theConfigManager.SetApplicationArgs(argc, argv);
6     if (theProcessHelper->startNotificationEngine() == SUCCESS
7         &&theProcessHelper->waitForBlackboardToStart() == SUCCESS
8         &&theMonitorManager.Init())

```

已进行静态页面的展示 (4.26)

Entities					Messages					Descriptors				
GBB	Entities	数量	最大	%	GBB	Message	数量	最大	%	GBB	Descriptor	数量	最大	%
1	1 Agent	0	100	0	1	51 Request...	0	100	0	1	69 Acousti...	0	100	0
2	2 SNAConn...	0	10	0	2	52 Allocat...	0	1000	0	2	172 ActiveS...	0	1000	0
3	51 IDAlloc...	0	1	0	3	53 ReleaseID	0	100	0	3	173 ActiveS...	0	1000	0
4	53 DBAreaE...	0	2000	0	4	55 Message...	0	100	0	4	171 ActiveS...	0	1000	0
5	54 Referen...	0	2000	0	5	56 PlayerC...	0	10	0	5	266 ActualT...	0	5000	0
6	56 DBRoute...	0	200	0	6	57 LoggerC...	0	10	0	6	1 AgentInfo	0	100	0
7	57 Referen...	0	200	0	7	59 NewRoute	0	100	0	7	2 AgentSt...	0	100	0
8	58 Mission	0	1	0	8	62 NewArea	0	100	0	8	309 Aggrega...	0	5000	0
9	59 Environ...	0	100	0	9	64 CreateE...	0	100	0	9	117 Airborn...	0	5000	0
10	60 RawEven...	0	1000	0	10	65 AddToMi...	0	100	0	10	242 Ammunit...	0	1000	0
11	61 Formatt...	0	1000	0	11	66 RemoveF...	0	100	0	11	57 AreaData	0	2000	0
12	62 UIForma...	0	1000	0	12	67 MPComma...	0	100	0	12	59 AreaDat...	0	200	0
13	63 Terrain...	0	100	0	13	68 SystemL...	0	10	0	13	224 AreasOf...	0	5000	0
14	66 Acousti...	0	10	0	14	69 DeleteE...	0	300	0	14	220 Attriti...	0	10000	0
15	67 ActiveS...	0	1000	0	15	72 AddEvent	0	100	0	15	306 AutoSca...	0	10	0
16	75 Chaff	0	500	0	16	74 AddDeco...	0	1000	0	16	305 AutoSca...	0	100	0
17	77 CommInt...	0	2000	0	17	75 Request...	0	100	0	17	255 Balloon...	0	500	0
18	80 Communi...	0	200	0	18	76 Allocat...	0	1000	0	18	86 BrainIn...	0	5000	0
19	81 Communi...	0	200	0	19	101 SetTime	0	100	0	19	85 BrainSt...	0	5000	0
20	82 Communi...	0	5000	0	20	102 SetOper...	0	100	0	20	217 C2BLTar...	0	5000	0
21	83 ECM	0	1000	0	21	103 Synchro...	0	100	0	21	273 C2Status	0	5000	0

在widget.cpp里对获得的数组vecinfo进行迭代展示代码如下，以entity为例

```

1 //初始化实例
2 StaticData staticdata;
3
4 staticdata.InitStructures();
5
6 //读取静态实体数据显示
7 staticdata.InitEntities();
8 for (int i = 0; i < staticdata.vecEntityInfo.size(); i++)
9 {
10     QString EnumType =
11     QString::number(staticdata.vecEntityInfo[i].EnumType);
12     QString EntityName =
13     QString::fromStdString(staticdata.vecEntityInfo[i].EntityName);
14     QString MaxEntityNum =
15     QString::number(staticdata.vecEntityInfo[i].MaxEntityNum);
16     model->setItem(i, 0, new QStandardItem(EnumType));

```

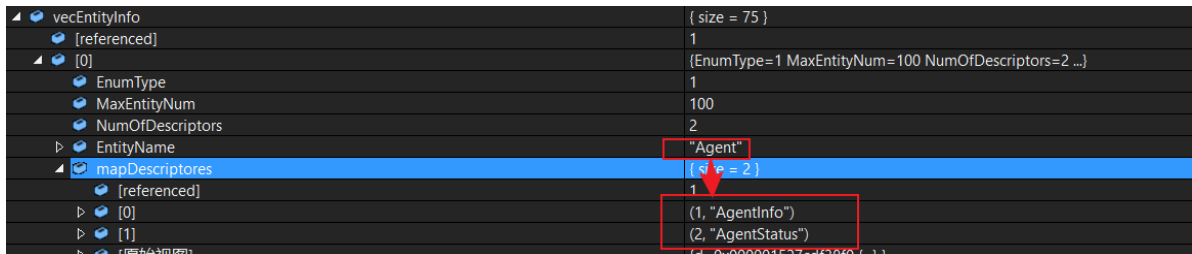
```

14         model->setItem(i, 1, new QStandardItem(EntityName));
15         model->setItem(i, 2, new QStandardItem("0"));
16         model->setItem(i, 3, new QStandardItem(MaxEntityNum));
17         //double rate = 0;
18         model->setItem(i, 4, new QStandardItem("0"));
19     }

```

各个数据关联情况

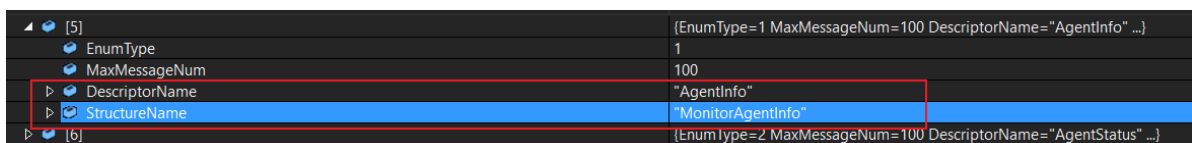
- Entity点击详情，以agent为例：



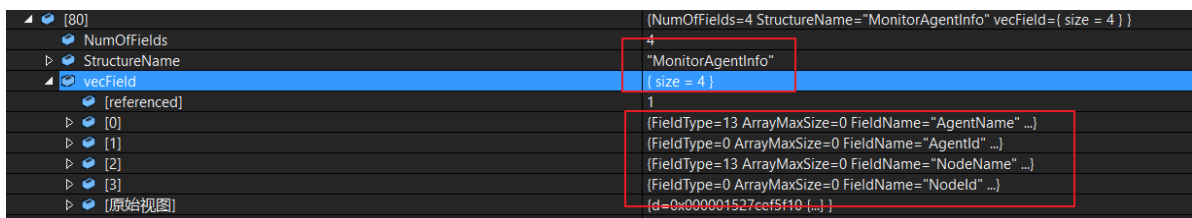
mapdescriptors是对应的2个根节点：



再去vecdescriptors中找到对应name的structure



根据 monitoragebtinfo 在vecstruct中找到子节点



- message详情，以requestnewid为例：

有的能在structname中找到，有的找不到

