

Teamcenter客户化开发（六）

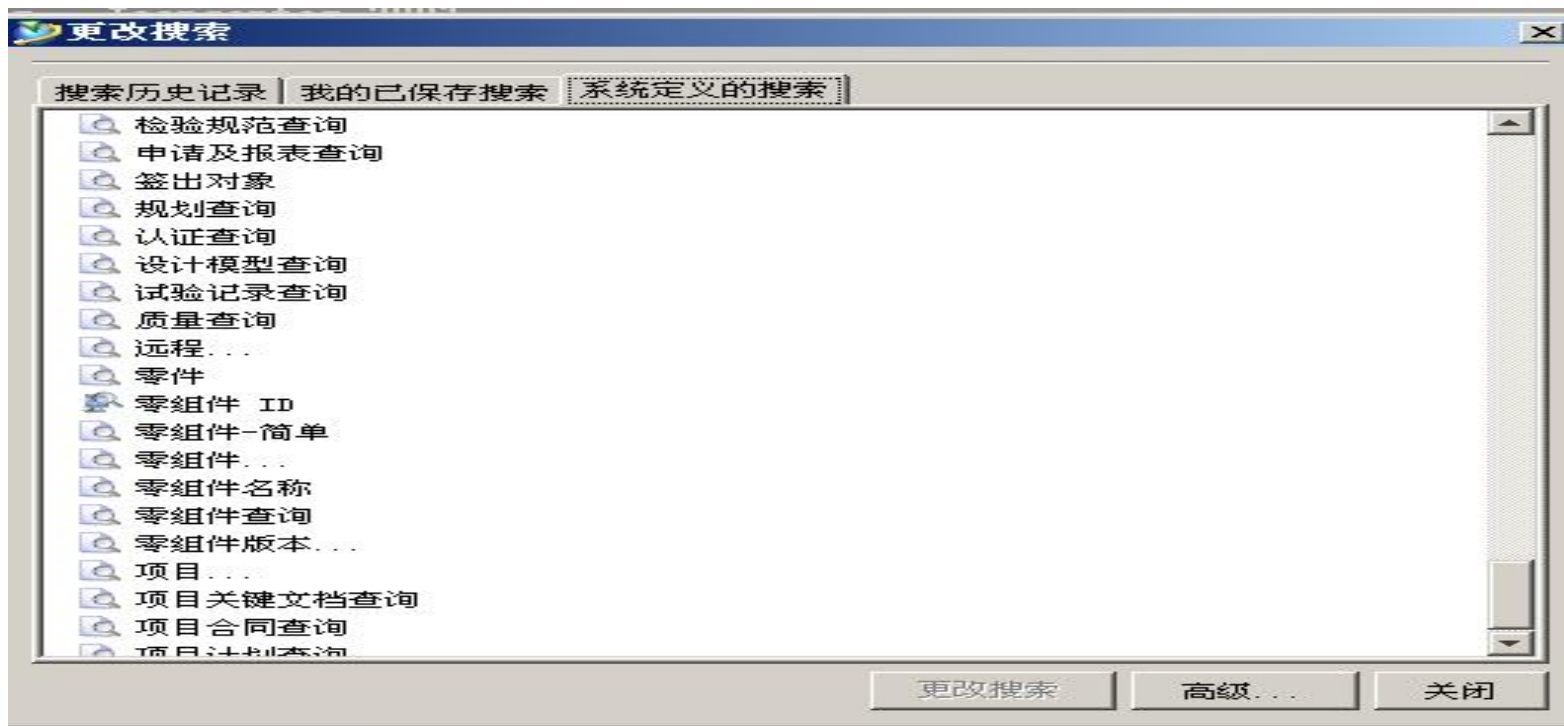
目录

Ø系统查询模块的使用

Ø报表设计

查询模块的调用

- Ø 在TC系统中，我们可以利用自定义的查询，查询系统中很多的数据。定义的每个查询都有自己的名称，那么我们在开发中就需要调用我们自己定义查询来查询数据
- Ø 系统中定义的查询：



查询调用方式

```
public static TCComponent[] query(TCSession session ,String query_name, String[] arg1, String[] arg2)
{
    TCComponentContextList imancomponentcontextlist = null;
    TCComponent[] component = (TCComponent[])null;
    try
    {
        TCComponentQueryType imancomponentquerytype =
(TCComponentQueryType)session.getTypeComponent("ImanQuery");
        TCComponentQuery imancomponentquery = (TCComponentQuery)imancomponentquerytype.find(query_name);
        TCTextService imantextservice = session.getTextService();
        String[] queryAttribute = new String[arg1.length];
        for (int i = 0; i < arg1.length; ++i)
            queryAttribute[i] = imantextservice.getTextValue(arg1[i]);

        String[] queryValues = new String[arg2.length];
        for (int i = 0; i < arg2.length; ++i)
            queryValues[i] = arg2[i];

        imancomponentcontextlist = imancomponentquery.getExecuteResultsList(queryAttribute, queryValues);
        component = imancomponentcontextlist.toTCComponentArray();
    }
    catch (Exception ex)
    {
        ex.printStackTrace();
    }
    return component;
}
```

Form displayed in a dialog box

TC中的报表

1.Java报表

2.Java+C报表

3.Java+SOA报表

JAVA报表

1. 此种方式是纯JAVA编写，通过各个对象的关系来查找数据，然后JAVA生成XLS，或者WORD的报表等各种报表。
2. 此方式适用于数据量小，数据提取的关系不复杂，采用此方式，速度可以让用户接受。
3. 关键点无。

JAVA+C编写

1. 此种方式是java+c编写，通过C直接在服务器执行，查找数据，然后在服务器共享目录生成一个文本或者其他格式的文件，在由客户端解析C生成的文件，由java来生成xls 或者word的报表等各种报表。
2. 此种报表适合数据量大，速度较快，比java执行速度估计快30%，现在一般采用此方法报表。
3. 关键点：因为是与用户交互型的开发，需要由java调用C，代码如下：

```
String str = session.getUserService().call(  
    "SERVICE_compare_bom_attrs", obj);
```

JAVA+SOA

1. 此种方式是java+soa编写，通过java 调用SOA执行，查找数据，然后在服务器先生成好数据，客户端在要调用报表命令的时候，直接调用已经生成完成的数据。然后生成xls 和 doc格式的数据
2. 此种报表适合数据量特大，速度比较慢，由于是要先成后好数据，其属于自动执行。缺点是数据是不是实时的。
3. 关键点：
S O A的数据查询：

```

public vector getallschedule( string queryname , string[] attribute , string[] valuse)
{
    vector v_schedule = new vector();
    imanquery query = null;
    savedqueryservice queryservice =
savedqueryservice.getservice(session.getconnection());
    datamanagementservice dataservice =
datamanagementservice.getservice(session.getconnection());
    queryservice.getclass();
    try {
        getsavedqueriesresponse savequery = queryservice.getsavesavedqueries();
        for (int i = 0; i < savequery.queries.length; i++) {
            if(savequery.queries[i].name.equals(queryname))
            {
                query = savequery.queries[i].query ;
                break ;
            }
        }

        savedqueryinput savedqueryinput[] = new savedqueryinput[1];
        savedqueryinput[0] = new savedqueryinput();
        savedqueryinput[0].query = query ;
        savedqueryinput[0].limitlist = new modelobject[0];

        savedqueryinput[0].entries = new string[attribute.length] ;
        savedqueryinput[0].values = new string[valuse.length];
        for (int i = 0; i < savedqueryinput[0].entries.length; i++) {
            savedqueryinput[0].entries[i] = attribute[i] ;
        }
    }
}

```

```

for (int i = 0; i < savedQueryInput[0].values.length; i++) {
    savedQueryInput[0].values[i] = valuse[i];
}
com.teamcenter.services.strong.query._2007_06.SavedQuery.ExecuteSavedQueriesResponse savedQueryResult
= queryService.executeSavedQueries(savedQueryInput);
SavedQueryResults found = savedQueryResult.arrayOfResults[0];

ModelObject[] object = found.objects;

for (int i = 0; i < object.length; i++) {
    try {
        Schedule sc = (Schedule) object[i];
        yq.loadobject(session,sc,"sch_summary_task");
        ModelObject schs =
sc.getProperty("sch_summary_task").getModelObjectValue();
        yq.loadobject(session,schs,"actual_finish_date");
        Calendar tpstrs =
schs.getProperty("actual_finish_date").getDateValue();

        if(tpstrs != null)
            continue;
        if(!v_Schedule.contains(sc))
        {
            v_Schedule.add(sc) ;
        }
    }
}

```

```
} catch (Exception e) {
```

```
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

```
    }  
} catch (ServiceException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}  
return v_Schedule;
```

```
}
```