# T24编程指南

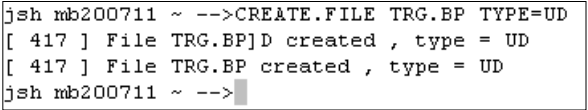
|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 描述 | 作者 |
| 0.1 | 2011-09-06 | 文档建立 | 郑敏嘉 |
| 0.2 | 2011-09-06 | Subroutine相关 | 郑敏嘉 |
|  |  |  |  |
|  |  |  |  |

## 简介

本文是对Temenos所给的编程文档进行总结，细化Temenos文档中描述过于简单的部分。对文档建立起一个清晰的结构，便于查阅。但并不是完整复制，内容过多需要查阅原文档的地方将会予以标注。

## 必备知识

### 创建目录



命令格式：CREATE.FILE FOLDERNAME TYPE=UD

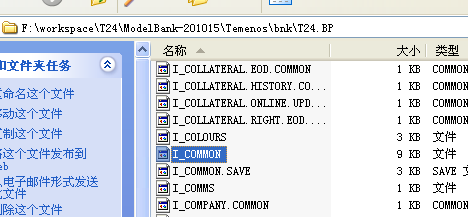
* FOLDERNAME ：要创建的目录名

### 编译

命令格式：BASIC [-I+INSFOLDER] PGMFOLDER PGMNAME

* INSFOLDER ：要引入的类库目录，比如要引入T24.BP, 则在BASIC 后加上-IT24.BP, 如要引入多个类库目录，可以跟多个“-I”

在workshop中，经常可以看到$INSERT I\_COMMON等引用，这里的I\_COMMON就在T24.BP下，如果不引入T24.BP，编译时会报I\_COMMON找不到



* PGMFOLDER ：源文件所在目录，该目录必须是在jsh下创建的目录
* PGMNAME ：文件名，文件名最好和程序同名，便于查找

*编译命令以%T24HOME%/bnk/bnk.run为根目录，执行编译之前，需要先启动remote.cmd，载入T24所需的环境变量。*

## 程序编写

这里理清一下T24编程的流程，结构，更多的语法、API要参阅

XIB-PRG1.Programming\_Using\_jBC-R10.pdf

XIB-PRG2.Programming\_Using\_T24\_APIs-R10.pdf

两份文档

### Program编程

留爪~

和subroutine查询文件名有差别，program必须要用FBNK.的前缀，subroutine用的是F.

### Subroutine编程

#### Subroutine说明



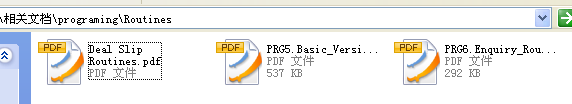
Subroutine是T24中，类似触发器的一个小程序，可以作为校验、取值、赋值等诸多用途，可以在Input/Authorise等各种条件下触发。在T24中，Subroutine有一定的规则，和java的接口相似，每一个Application有自己特有的Subroutine规则（一般是参数的个数不同），自定义Subroutine必须是遵守这个规则的一个实现，否则调用时将提示参数错误。这个参数作为输入或者输出变量，可以实现从交易内容中取值、修改等操作。

例如：

Version Routine没有参数

Enquiry Routine有一个参数, ENQ.DATA, 表示查询内容

每一个Routine的参数定义，要参照相应的Routine说明文档。



#### Subroutine的代码结构

SUBROUTINE ROUTINE.NAME(PARAMS1,PARAMS2…)

Statements

RETURN

END

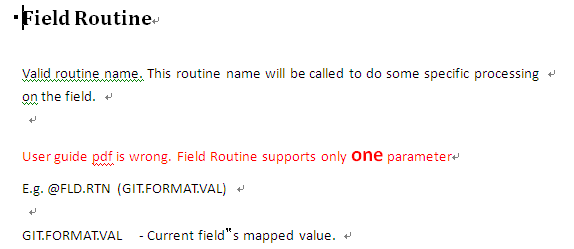
红字部分是固定格式，ROUTINE.NAME是SUBROUTINE的名称，参数的有无，数量多少，依底层Subroutine定义而定。Statements部分就是代码的主体部分，开发人员编写。

Subroutine没有返回值的概念，取而代之的是输入/输出参数，有可能PARAMS1作为输入，而处理结果需要赋值给PARAMS2，才能在UI上显示。

#### Subroutine编写、调用

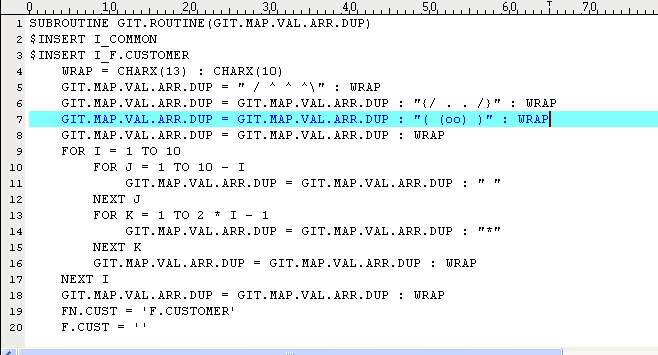
以一个GIT.FORMATTING.OUT Field Routine为例

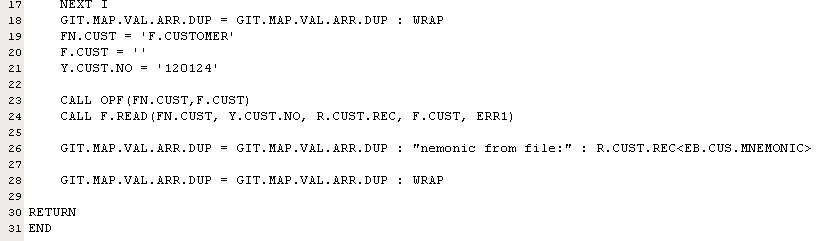
1. **首先，查找相关文档中关于Field Routine的定义。**



可以看到，这个Routine只有一个输入参数，这个输入参数的定义是：当前域的值映射。也就是说，在编程过程中，这个变量的值就是当前域的内容。可以获取或者对其进行修改。

1. 按照这个Routine的定义，可以新建一个源文件。这里演示了三个常用操作：字符串拼接、循环还有从数据库查询



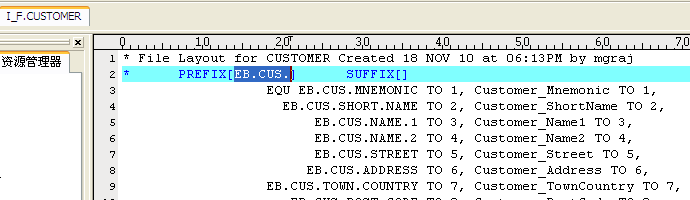


其中有几点需要注意

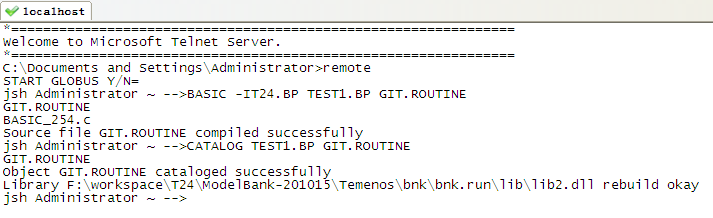
第一：两行$INSERT都是T24基础类，上面讲过，可以从T24.BP中引入。

第二：CHARX(13) :CHARX(10)是换行

第三：在查询操作中R.CUST.REC是查询结果，表示这是一条客户信息记录，R.CUST.REC<EB.CUS.MNEMONIC>是从客户信息中取MNEMONIC字段的值，这里的红字部分，不是自定义的，必须先查看类库文件中的定义，如下图所示。



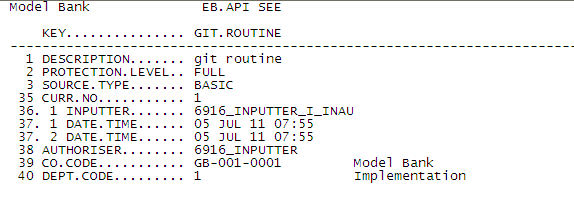
1. 进行编译和注册



CATALOG的时候，不用带-I的参数，在CATALOG之后，这个ROUTINE就会被注册到T24运行环境的动态链接库中，可以被识别，但是要调用还需要在EB.API中注册。

*Ps:反注册的命令是DECATALOG*

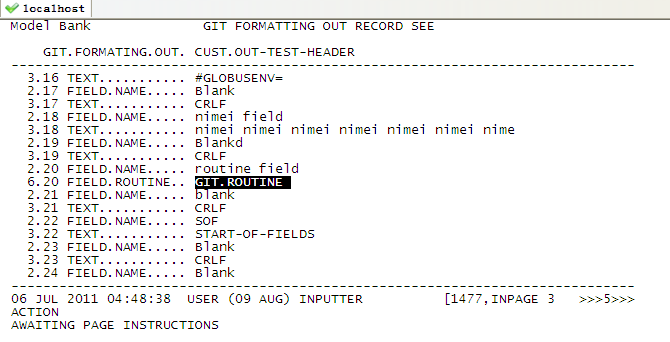
1. 在EB.API中注册



用classic或者browser都可以，新建一条记录，记录的ID必须和ROUTINE名称一致，PROTECTION LEVEL:FULL

SOURCE.TYPE: BASIC

之后，就可以把这个ROUTINE用于所需的地方。这里是用在GIT.FORMATTING.OUT的一个FIELD上



这个字段的值将由以上ROUTINE生成，最终输出效果如下

