

通过事件触发后台作业

Summary

Triggers an event in the background-processing system from an ABAP program.

Description

Events let you start background jobs under defined conditions. The event IDs are defined in transaction SM62 (event arguments are specified when the job is scheduled).

When you define a new event, a transport request must be manually created if it is to be transported to another system.

Parameters

EXPORTING
EVENTID The event name, defined in SM62
EVENTPARM Job can be scheduled to wait for an EVENTID or combination of EVENTID and EVENTPARM

Example

```
REPORT ZEXAMPLE.  
DATA: Q_EVENT LIKE TBTCJOB-EVENTID VALUE  'SAP_QEVENT' ,  
      Q_EVENTPARM LIKE TBTCJOB-EVENTPARM.  
  
CALL FUNCTION  'BP_EVENT_RAISE'  
EXPORTING  
  EVENTID = Q_EVENT  
  EVENTPARM = Q_EVENTPARM  
EXCEPTIONS  
  BAD_EVENTID = 1  
  EVENTID_DOES_NOT_EXIST = 2  
  EVENTID_MISSING = 3  
  RAISE_FAILED = 4  
  OTHERS = 5.  
IF SY-SUBRC NE 0.  
  WRITE:/  'EVENT' , Q_EVENT, 'NOT RAISED' .  
ELSE.  
  WRITE:/  'EVENT' , Q_EVENT, 'RAISED SUCCESSFULLY' .  
ENDIF.
```

1、利用system event 或者创建event. TCODE: SM62

显示 / 编辑事件

系统事件名

☒维护
☐显示

用户事件名

☐维护
☐显示

2、创建事件触发的后台作业

定义后台作业

开始条件 | 步骤 | Job selection | 个人作业 | 作业向导 | 标准作业

开始时间

立刻 | 日期/时间 | 作业后 | **事件后** | 在操作状态 | >..

日期/时间

作业后

在操作状态

事件后

事件 ZTEXT

参数 12

☒ 周期性的作业

检查

3、程序中调用函数 ‘BP_EVENT_RAISE’
当程序中该函数的事件与参数与job一致时，程序运行成功后机会触发后台作业运行。

【ABAP】使用Event来运行Background Job

使用Event来触发Background Job在SAP中应用也非常广泛, 另外在BW中是使用非常频繁, 最常见的一个应用就是使用Event来触发Process chain的执行(可以参考自己的BW PA培训笔记-ProcessChain Event来trigger).
在传统的ABAP中, 也许见得最多的就是使用Event来运行background job, 这个可以在SM36中进行定义.

1.实例

+ 1 - 刚开Custom Event

Tcode: SM62 Display/Edit Event
(如果是新版本, SM62与SM64均可; 同时还可以使用方法CREATE of class CL_BATCH_EVENT来创建)

Display / Edit Events

System event names ☐ Maintain ☐ Display

User event names ☐ Maintain ☒ Display

注意,这里是自己创建新的Event,也可以查看system/user event
然后可以得到目前存在的custom event

Edit User Events

Event identifier	Description
AP0_1	
AP0_2	
AP0_3	
CONVERT	
EWU_PHASEN_ALL_START_CLIENT	
EWU_PHASEN_START_CLIENT	
START_CC_IOR	
START_CC_I4R	
START_CC_I8R	
START_CONV	
ULOC	
ULOC_CLIENT_800	

这里点击create创建新的event

Create New Entry

Event

z_edison_job_1

Description

Trigger Background Job Event Test 1

Save

Cancel

保存即可.

+ 2 - 刚开Background Job的Step

Tcode: SE38

创建一个简单的report

```
*-----*
*& Report  ZTEST_JOB_1                               *
*&                                               *
*&                                               *
*&-----*
*& Report executed in Job                               *
*&                                               *
*&-----*

REPORT  ZTEST_JOB_1  no standard page heading  .

write:/ 'This is the report executed in Background job'.
```

+ 3 - 刚开Background Job

Tcode: SM36 Define Job

Define Background Job

Start condition

Step

Job selection

Own jobs

Job wizard

Standard jobs

General data

Job name

zevent_job_1

Job class

C

Status

Scheduled

Exec. Target

Spool list recipient

创建start condition

Start Time

Immediate

Date/Time

After job

After event

At operation mode

Date/Time

After job

After event

Event

Parameter

Periodic job

Check

Save

Cancel

Background Processing Event (3) 60 Entries Found

Restrictions

Background Processing Event

SAP_START_EHS_RDO_DISPATCHER

SAP_START_EHS_RDO_EVENTHANDLER

SAP_START_EHS_RDO_SAVE

SAP_START_WWI_DISPATCHER

SAP_START_WWI_WORKPROCESS

SAP_SYSTEM_START

SAP_SYSTEM_STOP

SAP_TEST

SAP_TRANSLATION_WORKLIST_START

SAP_TRIGGER_RDDIMPDP

SAP_TRIGGER_RDDIMPDP_CLIENT

SAP_TR_TXRP_DONE

START_CC_IOR

START_CC_I4R

START_CC_I8R

START_CONV

ULOC

ULOC_CLIENT_800

Z_EDISON_JOB_1

60 Entries Found

选择刚创建的Event

然后创建step即上面第二步所创建的report

最后Job为:

Define Background Job

Start condition Step Job selection Own jobs Job wizard Standard jobs

General data

Job name ZEVENT_JOB_1

Job class C

Status Scheduled

Exec. Target Spool list recipient

Job start

Scheduled start

After event Z_EDISON_JOB_1

Job frequency

Job steps

0000000001 -Steps successfully defined

然后save并release

+ 4 - 刚开Program桩Raise Event

Tcode: SE38

创建一个简单程序来Raise Event

```
*-----*
*& Report  ZTEST_JOB_2                                *
*&                                                *
*&-----*
*&   Raise Event to Execute Job                        *
*&                                                *
*&-----*

REPORT  ZTEST_JOB_2    no standard page heading  .

parameter: p_name  type BTCSEV-EVENTID default 'Z_EDISON_JOB_1'.

*---Raise Event

CALL FUNCTION 'BP_EVENT_RAISE'

EXPORTING

    EVENTID                = p_name
*   EVENTPARM              = ' '
*   TARGET_INSTANCE        = ' '

EXCEPTIONS

    BAD_EVENTID            = 1
    EVENTID_DOES_NOT_EXIST = 2
    EVENTID_MISSING        = 3
    RAISE_FAILED           = 4
    OTHERS                 = 5
.

IF SY-SUBRC <> 0.

    "Error

    message 'Raise Event Failed!' type 'E'.

else.

    "Successfully

    message 'Raise Event Successfully!' type 'S'.

ENDIF.
```

注意，BP_EVENT_RAISE已经obsolete了，可以使用cl_batch_event=>raise 来触发event. 如下

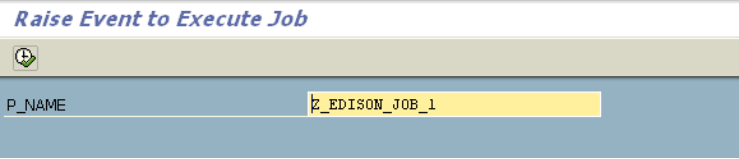
```
DATA:
  p_eventparm TYPE btcevtparm,
  p_server    TYPE btcserver.

parameter: p_name  type btceventid default 'Z_EDISON_JOB_1'.

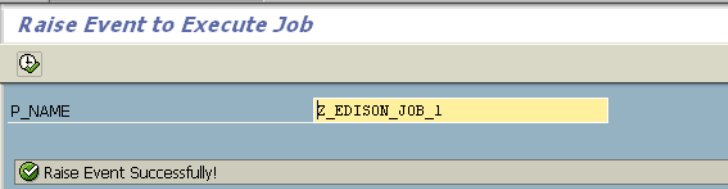
* Optional parameters: event parameter and target server.
p_eventparm = 'Event parameter'.
p_server    = ''.

* Raise Event
CALL METHOD cl_batch_event=>raise
EXPORTING
  i_eventparm      = p_eventparm
  i_server         = p_server
  i_eventid        = p_name
EXCEPTIONS
  except_raise_failed      = 1
  except_server_accepts_no_events = 2
  except_raise_forbidden   = 3
  except_unknown_event     = 4
  except_no_authority       = 5
  OTHERS                   = 6.
```

执行



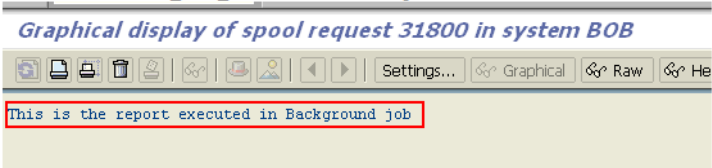
然后F8执行



然后去SM37中可以查看:

Event controlled ABAP program		Event ID: Program name :	
Job	Ln	Job CreatedBy	Status
EU_PUT		EDISON_TC	Finished
EU_REORG		EDISON_TC	Active
ZEVENT_JOB_1		EDISON_TC	Finished

可以发现ZEVENT_JOB_1被执行,可以看到spool list



使用Event触发Background job成功.

SM21 查看系统日志