

Oracle ERP最佳技术实践
E-BUSINESS SUITE
XX100 -应用技术
Business Events

Author: 黄建华Jianhua.Huang
MSN: huajhua@hotmail.com
Creation Date: September 19, 2006
Last Updated: December 14, 2006
Document Ref: <Document Reference Number>
Version: DRAFT 1A

Approvals:

<Approver 1>

<Approver 2>

Document Control

Change Record

Date	Author	Version	Change Reference
12-Apr-05	Jianhua.Huang	Draft 1a	No Previous Document

Reviewers

Name	Position

Distribution

Copy No.	Name	Location
1	Library Master	Project Library
2		Project Manager
3		
4		

Note To Holders:

If you receive an electronic copy of this document and print it out, please write your name on the equivalent of the cover page, for document control purposes.

If you receive a hard copy of this document, please write your name on the front cover, for document control purposes.

Contents

Document Control	2
1. Business Events System基础	4
1.1. 总体说明	4
1.2. 应用举例	5
1.3. 必要设置	5
1.4. 先睹为快: 一个简单例子.定义测试表	6
1.5. 先睹为快: 一个简单例子.定义事件和订阅关系	7
1.6. 先睹为快: 一个简单例子.立即执行	11
2. 其他特性和功能实例	13
2.1. 关键概念理解	13
2.2. 关于Defer	17
2.3. 第2个例子: Defer执行	18
2.4. 第3个例子: 发送Notification	19
2.5. 第4个例子: 创建Agent Listener	21
2.6. 第5个例子: Workflow产生/发送消息	23
2.7. 第6个例子: Workflow接收消息	27
3. 常用代码块	31
3.1. 查看订阅关系	31
3.2. 启动事件	31
3.3. 发送事件到特定Agent	32
3.4. 查看事件	32
4. Open and Closed Issues for this Deliverable	33
Open Issues	33
Closed Issues	33

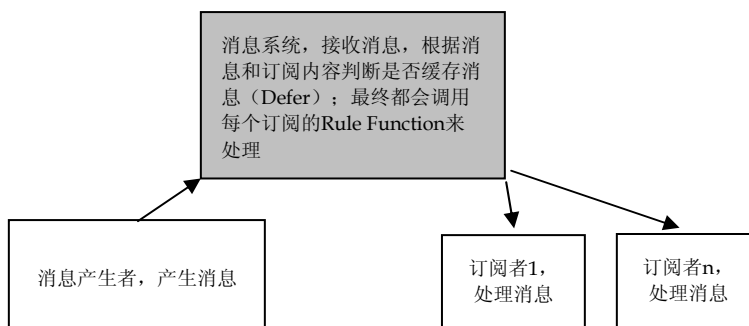
1. Business Events System基础

职责: Workflow Administrator Web Applications (以下简称WF Admin Web)

用户: SYSADMIN

1.1. 总体说明

Oracle业务时间系统，就是一个消息产生和消费的过程，Oracle打算用它来做系统集成。源系统产生数据，封装成业务事件，发送给目标系统，目标系统做出处理响应。这里的源系统和目标系统，可能是ERP中的不同模块，也可能是两个ERP Instance。



佚名: “Business Event可能在APPLICATION的集成方面大有作为”!

业务事件是以AQ为基础，如果能够明白AQ的工作原理，就可以更好的理解BES了。关于AQ，请参考附件中的《AQ快速入门.txt》。

胡永福曾写过一个精彩的实例文档《深入浅出OracleEBS 11i-BES的一个简单应用》，描述如何通过AQ+BES+Workflow处理外挂系统和ERP的信息交互，心急的朋友可以直接去阅读。

Business Events丰富和扩展了DB标准AQ的功能，但基本原理还是一样的。

EBS中用的比较多的仍旧是简单的消息订阅模式，即某个功能产生数据后，不直接更新/同步到其他功能块，而是Raise一个消息到队列，然后立即返回操作界面；待后台的Listener起来后，根据订阅关系执行相关操作。

这里面的优点是：前端操作不在同一Transaction处理太多事情，性能得到改善；仅通过消息传递，功能块之间的耦合度小；对于不同系统间的集成操作，提供了非常成熟的方案。

不过还是觉得不够干脆：比如不同系统间，还是DB_LINK来得直接，没有中间处理，效率也不错；本系统内部，用了Business，徒增一个动作的执行路径，搞不好还容易导致问题。

1.2. 应用举例

处理问题

EBS中广泛使用BES，如果不了解，遇到问题岂不麻烦？

William Shi在《关于用户/职责方面的修改》中处理的关键问题就是BES引发的问题。

系统集成

胡永福的《深入浅出OracleEBS 11i-BES的一个简单应用》。

操作监控

Oracle的Alert具有监控功能，但其基于DB触发器，可能引起性能问题。

如果采用BES，在用操作时记录增删改Event，再Defer到后台，统一处理。系统中有些标准功能如BOM，已经有此类事件产生，我们仅需做订阅即可达到监控目的。

1.3. 必要设置

数据库参数

BES本质上使用AQ，AQ本质上使用数据库JOB。所以要保证AQ和Job工作正常。通常建议：

```
job_queue_processes    >=10
aq_tm_processes        >=1
```

```
SELECT p.NAME, p.description, p.VALUE
FROM v$parameter p
WHERE p.NAME IN ('job_queue_processes', 'aq_tm_processes')
```

Profile: Concurrent:GSM Enabled

在Site层设置为Yes。

```
SELECT pro.profile_option_name,
       pro.user_profile_option_name,
       prv.profile_option_value
FROM apps.fnd_profile_options_vl pro,
     applsys.fnd_profile_option_values prv
WHERE pro.profile_option_id = prv.profile_option_id(+)
      AND prv.level_id = 10001
      AND pro.profile_option_name = 'CONC_GSM_ENABLED'
```

Concurrent Manager: Service Manager: <Host_Name>

Target和Actual个数要>=1。如Service Manager: GTHF。

```
SELECT fcq.concurrent_queue_name,
       fcq.user_concurrent_queue_name,
       fcq.target_node,
       fcq.max_processes target,
       fcq.running_processes actual
FROM fnd_concurrent_queues_vl fcq, v$instance ins
```

```
WHERE fcq.enabled_flag = 'Y'  
AND fcq.concurrent_queue_name = 'FNDSM_' || upper(ins.host_name)
```

Concurrent Manager: Workflow Agent Listener Service

Target和Actual个数要>=1。

```
SELECT fcq.concurrent_queue_name,  
       fcq.user_concurrent_queue_name,  
       fcq.target_node,  
       fcq.max_processes target,  
       fcq.running_processes actual,  
       fcqs.sleep_seconds  
FROM   fnd_concurrent_queues_vl fcq, fnd_concurrent_queue_size fcqs  
WHERE  fcq.concurrent_queue_id = fcqs.concurrent_queue_id  
AND    fcq.enabled_flag = 'Y'  
AND    fcq.concurrent_queue_name = 'WFALSNRSVC'
```

上面3步设置，摘自William Shi的“关于用户/职责方面的修改”：

怎样让Business Event系统保持的工作

- 1、首先，我们必须保证一个叫做“Workflow Agent Listener Service”的Concurrent Manager正常工作，这个可以通过SYSADMIN职责的“Administer Concurrent Manager”Form来看，要保证它的Target=1
- 2、如果你怎么整也没办法把这个Target整成1，那么你需要检查下一个称为“Service Manager: <hostname>”的Concurrent Manager是否正常工作，也就是说，它的Target必须先为1
- 3、如果这个东东也整来整去整不成1，那么你需要检查一个“Concurrent:GSM Enabled”的Profile，是不是在Site层设为“Y”的，如果是“N”的话，也是不行的

Workflow必要设置，如果打算在WF中使用BES

参考《Oracle E-Business Suite 实例详解之Workflow.doc》之“1.2. 必要设置”。

1.4. 先睹为快：一个简单例子.定义测试表

准备数据库表

创建表：

```
-- Create table  
create table CUX_BES_DEMOS  
(  
  DEMO_ID          NUMBER,  
  MESSAGE_NAME    VARCHAR2(100),  
  MESSAGE_KEY     VARCHAR2(100),  
  MESSAGE_CONTENTS CLOB,  
  PARAMETER_VALUE1 VARCHAR2(4000),  
  PARAMETER_VALUE2 VARCHAR2(4000),  
  EVENT_DATE      DATE  
)  
  
CREATE SEQUENCE CUX_BES_DEMO_S;
```

准备数据库Package

注意参数和返回值，要按约定写，而且不能有Commit语句。创建Package：

```
CREATE OR REPLACE PACKAGE cux_bes_demo AUTHID CURRENT_USER AS  
  --Created by Jianhua.huang 2006.09.19
```

```

FUNCTION process_an_event(p_sub_guid IN RAW,
                        p_event      IN OUT NOCOPY wf_event_t)
RETURN VARCHAR2;
END;
/

CREATE OR REPLACE PACKAGE BODY cux_bes_demo AS
FUNCTION process_an_event(p_sub_guid IN RAW,
                        p_event      IN OUT NOCOPY wf_event_t)
RETURN VARCHAR2 IS

l_demo_rec cux_bes_demos%ROWTYPE;

BEGIN

SELECT cux_bes_demo_s.NEXTVAL INTO l_demo_rec.demo_id FROM dual;

--get evevent itself
l_demo_rec.message_name := p_event.geteventname;
l_demo_rec.event_date   := p_event.getsenddate;

--we can use this key to transfer message only
l_demo_rec.message_key := p_event.geteventkey;

--optional, get xml data
l_demo_rec.message_contents := p_event.geteventdata;

--optional, get parameter
l_demo_rec.parameter_value1 := p_event.getvalueforparameter('PARAMETER1');
l_demo_rec.parameter_value2 := p_event.getvalueforparameter('PARAMETER2');

--do anything here
INSERT INTO cux_bes_demos cbd VALUES l_demo_rec;

RETURN 'SUCCESS';
EXCEPTION
WHEN OTHERS THEN
RETURN 'ERROR';
END;
END;

```

1.5. 先睹为快：一个简单例子.定义事件和订阅关系

注册Event

N: WF Admin Web/Business Events/Create Event

定义如下：

字段	值	说明
Name	cux.business.events.demo01	必须
Display Name	Business Events Demo01	必须
Status	Enabled	必须

字段	值	说明
Generate Function		一个消息产生时，除了Name、Key，还可以有XML格式的Data，有些Subscription的处理者需要，有些不需要。如果需要（Rule Data=Message），并且消息产生时没有给Data，BES就会调用这里的函数来产生，当然这个函数是要遵照特定的参数和返回类型；如果放空，系统就用默认值填充。可参考wf_event.setmessage
Java Generate Function		上述函数Java版
Owner Name	CUX	必须，Program Name
Owner Tag	CUX	必须，Program ID
Customization Level	User	必须，自定义的都是User Core 系统自带，不能更改 Limit 系统自带，仅可失效/启用

注意上面的事件名不规范哦：

Name约定格式为<company>.<family>.<product>.<component>.<object>.<event>

* Name

* Display Name

Description

* Status

Generate Function

Java Generate Function

* Owner Name

* Owner Tag

Customization Level


```
SELECT *
FROM wf_events_vl wfe
WHERE wfe.NAME = 'cux.business.events.demo01'
```

定义Subscription，注意这里的Action Type

N: WF Admin Web/Business Events/Subscription

把刚才定义的事件查询出来：

The screenshot shows the Oracle Business Events web interface. At the top, there are navigation links: Events | Subscriptions | Agents | Systems. Below that, the page title is 'Business Events: Events'. A search bar contains the text 'cux.business.events.demo01' and a 'Go' button. Below the search bar, there is a 'Results: Events' section with a table. The table has columns: Select Name /, Display Name, Type, Status, Subscription, Update, and Test. The first row in the table is: cux.business.events.demo01, Business Events Demo01, Event, Enabled, [Subscription icon], [Update icon], [Test icon]. At the bottom of the page, there is a footer with copyright information and a 'Privacy Statement' link.

点击Subscription列的图标，在导航到的新界面点击  :

字段	值	说明
基本信息		
System	GTPROD.WORLD	必须
Source Type	Local	必须 Local: to events raised on the subscribing system External: to events received by an inbound agent Error: to errored events dequeued from the WF_ERROR queue.
Event Filter	cux.business.events.demo01	必须
Source Agent		意思是从哪个Agent来的消息才是本订阅要的。一般放空，代表所有
Phase	90	如果一个事件有多个订阅，那么按此顺序执行。此外，1- 99同步执行，即立即执行，用这个测试比较方便；100及以上Defer，也就是异步，从AQ角度看，消息先入列
Status	Enabled	必须
Rule Data	Message	必须。 Message需要准备XML Data Key不需要
Action Type	Custom	必须，类型很多，可以顾名思义： Custom Launch Workflow Receive Trading Partner Message Send Notification Send Trading Partner Message Send to Agent
On Error	Stop and Rollback	必须，遇到错误怎么办 Stop and Rollback回滚所有订阅，不再继续 Skip to Next 回滚出错的订阅，继续执行下一个订阅
Action		
.....		
PL/SQL Rule Function	cux_bes_demo.process_an_event	处理函数。 除了实现自定义功能，我们可以在自己编写的函数中调用系统标准的函数（wf_rule.xx），以达到上述Action Type中的其他功能。
Java Rule Function		Java版处理函数
.....		
Parameter 这里定义的参数，可能有2个作用，一是加入Event消息内容的Parameter List中，二是和Event消息内容已有的Parameter List进行比较。这些要看具体的订阅如何定义，比如Launch Workflow中的2个选项就是分别这样用的。		
Parameter1	Value1	可选
Parameter2	Value2	可选
.....		
Documentation		
Owner Name	CUX	必须
Owner Tag	CUX	必须
.....		

具体屏幕步骤：

Create Event Subscription

An event subscription is a registration indicating that a particular event is significant to a particular system. An event subscription specifies the processing to perform when the triggering event occurs.
* Indicates required field

Subscriber
* System

Triggering Event
* Source Type
* Event Filter
Source Agent

Execution Condition
* Phase
Subscription with a phase 1-99 are run synchronously , 100 and above are deferred.
* Status
* Rule Data

Action Type
* Action Type
The Action Type controls the behavior of the subscription
On Error

Action Type	Description
Custom	Execute custom business logic
Launch Workflow	Launch the workflow specified in the subscription
Receive Trading Partner Message	Receive a message from your trading partner
Send Notification	Send a notification using standard or your own message templates
Send Trading Partner Message	Generate a message and send the message to your trading partner
Send to Agent	Send the event to the agent specified in the subscription

Next

Create Event Subscription: Custom Subscription

Action
The Rule Function controls the behavior of the subscription. Provide a Java Class name (<Package> <Class>) for Java Rule Function and a PLUSQL stored procedure (<Package> <Function>) for PLUSQL Rule Function.

Java Rule Function
PLUSQL Rule Function
Workflow Type
Workflow Process
Choose a Workflow Type, before choosing the Workflow Process for that Type
Out Agent
To Agent
* Priority

Subscription Parameters
Select Object:
Select All | Select None
Select Name | Value

<input type="checkbox"/>	PARAMETER1	VALUE1
<input type="checkbox"/>	PARAMETER2	VALUE2

Enter parameters and their values with no spaces

Documentation
* Owner Name
* Owner Tag
Customization Level
Description

Home | Logout | Preferences | Help

```
SELECT sub.guid sub_guid,
       decode(sub.rule_function,
              NULL,
              decode(sub.java_rule_func,
                     NULL,
                     NULL,
                     'java://' || sub.java_rule_func),
              sub.rule_function) rule_func,
       sub.source_type source_type,
       nvl(sub.phase, 0) phase,
       sub.rule_data rule_data,
       sub.priority priority,
       sub.on_error_code on_error
FROM wf_event_subscriptions sub, wf_events evt
WHERE sub.event_filter_guid = evt.guid
AND evt.NAME IN ('cux.business.events.demo01');
```

1.6. 先睹为快：一个简单例子.立即执行

测试1：界面测试

N: WF Admin Web/Business Events/Test

把刚才定义的cux.business.events.demo01事件查询出来：

Select Name /	Display Name	Type	Status	Subscription	Update	Test
<input type="checkbox"/> cux.business.events.demo01	Business Events Demo01	Event	Enabled			

点击Test列的图标

输入如下信息，参数和XML Data是可选的：

Test Business Event
To test a business event enter an event key , a list of parameter name / value pairs (optional), and either paste in an XML Document or upload from the local file system and press submit. Cancel Submit

* Indicates required field

Event Identifier

* Event Name

* Event Key

Send Date
Send Date must be in the Format: CC-MON-RRRR

Event Parameters

Select Object: Delete

Select All | Select None

Select Label	Value
<input type="checkbox"/> PARAMETER1	Jianhua Huang

Add Another Row

Event Data

XML Content

If Xml content is greater than 4000 char , Press "Upload XML File" Button.

Upload Option Upload XML Write XML File Name < No File Chosen to Upload >

Upload Xml File Cancel Submit

Submit后提示成功：

Events | Subscriptions | Agents | Systems

Information

Information - The Event <cux.business.events.demo01> has been raised.

Test Business Event

用SQL查询：

```
select * from cux_bes_demos t;
```

DEMO_ID	MESSAGE_NAME	MESSAGE_KEY	MESSAGE_CONTENTS	PARAMETER_VALUE1	PARAMETER_VALUE2	EVENT_DATE
1	cux.business.events.demo01	just_test01	<CLOB>	Jianhua.Huang		2006-9-20 13:22:08

可以看到，数据已经进入表；而实际上，我们可以做更多的处理。

测试2：脚本测试

也可以用脚本触发事件：

```
DECLARE  
  x_parameters wf_parameter_list_t;  
  l_event_data CLOB;  
BEGIN
```

```

-- Call the procedure
-- Set Parameter
wf_event.addparametertolist(p name      => 'PARAMETER1',
                           p value     => 'Jianhua',
                           p_parameterlist => x_parameters);
wf_event.addparametertolist(p_name     => 'PARAMETER2',
                           p_value     => 'Huang',
                           p_parameterlist => x_parameters);

l_event_data := '<?xml version="1.0" encoding="UTF-8"?><root><Name>Jianhua.Huang</Name></root>';

wf_event.RAISE(p_event_name => 'cux.business.events.demo01',
               p_event_key  => 'just test:02',
               p_event_data => l_event_data,
               p_parameters => x_parameters,
               p_send_date  => SYSDATE);
END;

```

结果一样:

DEMO_ID	MESSAGE_NAME	MESSAGE_KEY	MESSAGE_CONTENTS	PARAMETER_VALUE1	PARAMETER_VALUE2	EVENT_DATE
1	cux.business.events.demo01	just_test:01	<CLOB>	Jianhua.Huang		2006-9-20 13:22:08
2	cux.business.events.demo01	just_test:02	<CLOB>	Jianhua	Huang	2006-9-20 13:22:56

2. 其他特性和功能实例

上面的例子虽然极其简单，但演示了整个过程，至少可以初学者迈出第一步；接下来我们看看一些其他重要的功能，至于基本操作，就不再详细说明了。

2.1. 关键概念理解

Event Manager

即业务事件系统模块，管理本节介绍的所有对象：System、Agent&Group、Event&Group、Subscription。当然，其本身也是个处理Engine，有一套API如wf_events等，一系列预定义的事件和Agent。

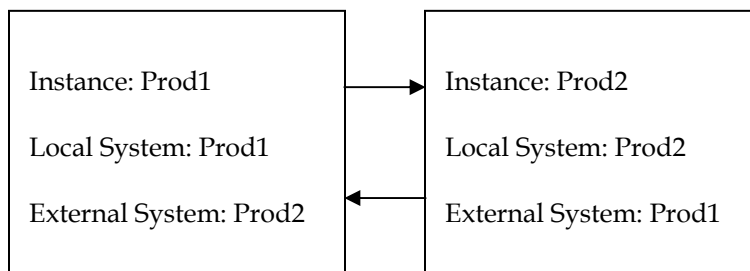
System

类似ERP中Node的定义，System代表一台主机或者一个DB实例，其核心意义是一个“事件系统”：ERP系统本身、或者一个外围系统。

Local System: 只要装上了Workflow，就会自动安装一个以DB Global Name命名的系统，代表ERP系统本身，也叫“Local System”。虽然在Workflow的全局配置页面可以更改，但100%没必要，所以这个也相当于“内置”System了。注意这里仅仅在名字上叫Local，与Subscription中的消息来源Local是两码事。

External System: 相对于Local System，其他任何新定义的系统都是外部系统。

Local和External是相对的，另外一个叫法是Source和Destination。假如有两个Instance叫Prod1、Prod2，如果需要在这两个系统间传递消息，那么需要互相定义：



如果Prod1和Prod2都是Workflow系统，那么定义相对比较简单，可以在Prod2导出Local System定义（包含Agent等）到一个XML文件，在Prod1作为External System导入；反之亦然。

有趣的是，这里的导入也是通过在Prod1触发一个叫“System Signup”的事件并以Prod2导出的XML文件作为Data来实现的。

Tips: 另外，以后修改了定义，就可以直接在这两个系统间通过一个叫“Synchronize Event Systems”的事件来同步了；甚至可以设置自动同步。

消息结构, AQ的Payload, Queue Handler

一个具体的消息, 包含事件名、消息关键字、发送时间、可选的发送数据, 在Workflow系统中用一个标准对象来表示: **WF_EVENT_T**, 对应的Java类为BusinessEvent。

一个AQ队列, 都有一个对象类型作为有效负载(Payload)。

Queue Handler: 封装DBMS_AQ的enqueue和dequeue, 供其他程序调用; 当然这个封装未必简单, 可能需要在不同消息结构和AQ的Payload之间进行转换。

BES默认的Defer消息队列wf_deferred, 其有效负载也是WF_EVENT_T, 所以标准的Queue Handler中无需转换。

消息产生、返回值、事务

顾名思义, Event Group可包含一组Event, 从而可以简单的对组进行订阅。

消息产生时, 必须指定是哪个Event, 不能针对Group。可以有多种方式产生:

- 1、前面例子的raise函数
- 2、Workflow的Event Activity
- 3、AQ的消息入列
- 4、手工在Business Event页面提交, 直接输入XML Data限制为4000字节, 上传附件XML Data则无限制

关于返回值: 实际上, Raise事件时也可以获得一些返回值, 这可以通过改用WF_EVENT.Raise3来实现, Raise3和Raise的不同点在于Raise3的Event参数是In Out的。

关于事务: Deferrer的Event, 触发者和Subscription处于不同的Transaction; 非Deferrer的Event, 触发者和Subscription处于同一个Transaction。

消息订阅

*这里的订阅和标准AQ的订阅有所不同, 标准AQ订阅指从一个队列, 按照一定Where条件分发到其他订阅者队列, 它们具有相同的payload结构。

这里消息订阅的意思就是, 当某个消息产生后, 要通知我处理, 并把消息内容传递给我, 至于怎么处理, 由订阅者自己来搞。

订阅可以针对组或者事件。如果是针对组, 组中任意一个事件产生消息时, 都会触发订阅Action的执行。

同一个消息的订阅按照定义的Phase顺序执行, 前面的订约如果修改了Event内容, 则会回传给下一个订阅者。合理使用Phase和On Error, 可以做到前面的订阅者负责验证数据, 后面的订阅者负责实际处理数据。

一个消息如果没有合适的订阅者, 则会交给处理Unexpected event的订阅者处理。

合适的订阅者: 有效的Local System订阅者, 并且消息名、Source Type和消息一致。

Agent

不同的事件，其发生频率不同、分发规则不同、发送方向不同、甚至协议和有效负载（Payload）的格式也不同，最好分开管理；BES中用Agent来分类管理。

Agent共同作用：Agent仅仅是一个定义，定义如何管理一个AQ对列，主要内容是协议和Queue Handler；BES接收到Defer消息，调用Agent的Queue Handler（QH）入列；BES发现队列有消息，调用Agent的QH出列。当然了，也可以利用DBMS_AQ，在底层直接操作队列。

Inbound Agent：对本系统来说，队列是用来接收消息的，消息源（任意系统）最好调用该Agent定义的QH把消息入列；Inbound Agent是需要建立Listener侦听的，其侦听到消息，调用Agent的QH出列，再调用BES API进一步处理订阅。

Outbound Agent：对本系统来说，队列是用来发送消息的，消息源（本系统）一般调用该Agent定义的QH把消息入列；Outbound Agent是需要建立Propagation分发的，和侦听类似，发现队列中有消息，调用Agent的QH出列，根据分发规则，进一步调用目标系统Inbound Agent的QH把消息入列。

Protocol：就是协议，目前EBS支持HTTP、SMTP、SQLNET。系统支持创建客户化的协议，但通常我们仅涉及SQLNET。

```
SELECT * FROM wf_lookups WHERE lookup_type = 'WF_AQ_PROTOCOLS'
```

SQLNET协议：使用WF_EVENT_T作为消息结构，默认使用标准的Queue Handler包，WF_EVENT_QH、WF_ERROR_QH，来处理AQ的入列/出列操作。

Agent Listener

因为Agent已经定义了如何管理AQ对列，进一步可以通过简单的定义，由BES在运行时组成动态SQL来实现AQ Listener，而不用如《AQ快速入门.txt》介绍的那样，自己写Listener的代码了。这就是Agent Listener！如下是系统的一个定义实例：

Queue

Name:	WF_DEFERRED
Payload:	WF_EVENT_T

Agent

Name:	WF_DEFERRED
Protocol:	SQLNET
Queue Handler:	WF_EVENT_QH
Direction:	IN

Agent Listener

Name:	Workflow Deferred Agent Listener
Sleep:	120秒

1、消息产生：源系统Raise消息，如Defer则调用WF_EVENT_QH.enqueue，压入WF_DEFERRED对列。

2、消息处理：Workflow Deferred Agent Listener，按120秒的间隔侦听WF_DEFERRED对列，如有消息到期则用WF_EVENT_QH.dequeue出列，得到WF_EVENT_T实例，然后调用BES的API执行事件相应的Subscription代码。

ERP默认安装：

WF_DEFERRED	Workflow Deferred Agent Listener
WF_DEFERRED	Workflow Deferred Notification Agent Listener
WF_ERROR	Workflow Error Agent Listener
WF_NOTIFICATION_IN	Workflow Inbound Notifications Agent Listener
.....	

\$FND_TOP/wfagtlst.sql可以用来启动一个Agent的Listener，也可以在Workflow Manager的管理界面增加。

多个Listener同时处理一个Agent，会冲突否？

谁启动Agent Listener，何时启动

不同的消息队列由不同的Agent Listener处理。WF_DEFERRED队列由Workflow Deferred Agent Listener处理，而其是由并发管理器Workflow Agent Listener Service启动的。本质还是启动AQ的Listen。

Workflow Agent Listener Service是一个Generic Service Management (GSM) 服务。

Select Name	Status	Type	Startup Mode	Container Type	Container	Actions
ECX Inbound Agent Listener	Stopped	Workflow Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
ECX Transaction Agent Listener	Stopped	Workflow Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Web Services IN Agent	Stopped	Workflow Java Agent Listener	Manual	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Web Services OUT Agent	Stopped	Workflow Web Services Outbound	Manual	Oracle Applications GSM	Workflow Document Web Services Service	Refresh Go
Workflow Deferred Agent Listener	Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Deferred Notification Agent Listener	Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Error Agent Listener	Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Inbound Notifications Agent Listener	Running	Workflow Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Java Deferred Agent Listener	System Deactivated	Workflow Java Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Java Error Agent Listener	Running	Workflow Java Agent Listener	Automatic	Oracle Applications GSM	Workflow Agent Listener Service	Refresh Go
Workflow Notification Mailer	Stopped	Workflow Mailer	Automatic	Oracle Applications GSM	Workflow Mailer Service	Refresh Go

Propagation

和Listen一样，Propagation也是AQ的标准功能。前者依赖AQ Time Managers，后者以来Job Queue，对应数据库参数分别是aq_tm_processes、job_queue_processes。

《AQ快速入门.txt》有完整的Propagation简单例子。

目前还没发现ERP中哪里用了Propagation，这里就暂且不再深入说明。

License

自定义的事件，都是Licensed，只要有消息产生，就会促发订阅的执行；如果是EBS Seeded的，则要看相应的模块是否Licensed，否则不执行，事件界面也会显示event is not licensed。

通过OAM页面增加某个模块的License后，需要运行请求**Synchronize Product License and Workflow BES License**。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/987163125120006052>