Introducing Oracle Queuing/Messaging Technology

Anthony D. Noriega
MSCS, MBA, BSSE, OCP-DBA

ORACLE | CERTIFIED PROFESSIONAL

Objectives

- Emphasize technical concepts and Oracle queuing infrastructure technology.
- Highlight programming techniques, methodologies, and relevant architectures.
- Discuss the relevant of queuing business models in various industries, with emphasis on industry and business requirements.

Objectives

- Emphasize the reliability of inter process communication imposed by AQ technology.
- Position Oracle as the de facto leader in the integrated database/advanced message queuing market.

Technical Concepts

- Queue (FIFO data structure)
- Message Queuing
- Header
- Payload
- Channel
- Port
- Propagation

Technical Concepts

- Producer (enqueuing)
- Consumer (dequeuing)
- Recipient
- Enqueue
- Dequeue

Technical Concepts

- Peer-to-Peer Mode
- Publish/Subscribe Mode
  - Broadcasting
  - Multicasting
- Streams AQ
- Model View Controller (Message-Driven Beans)

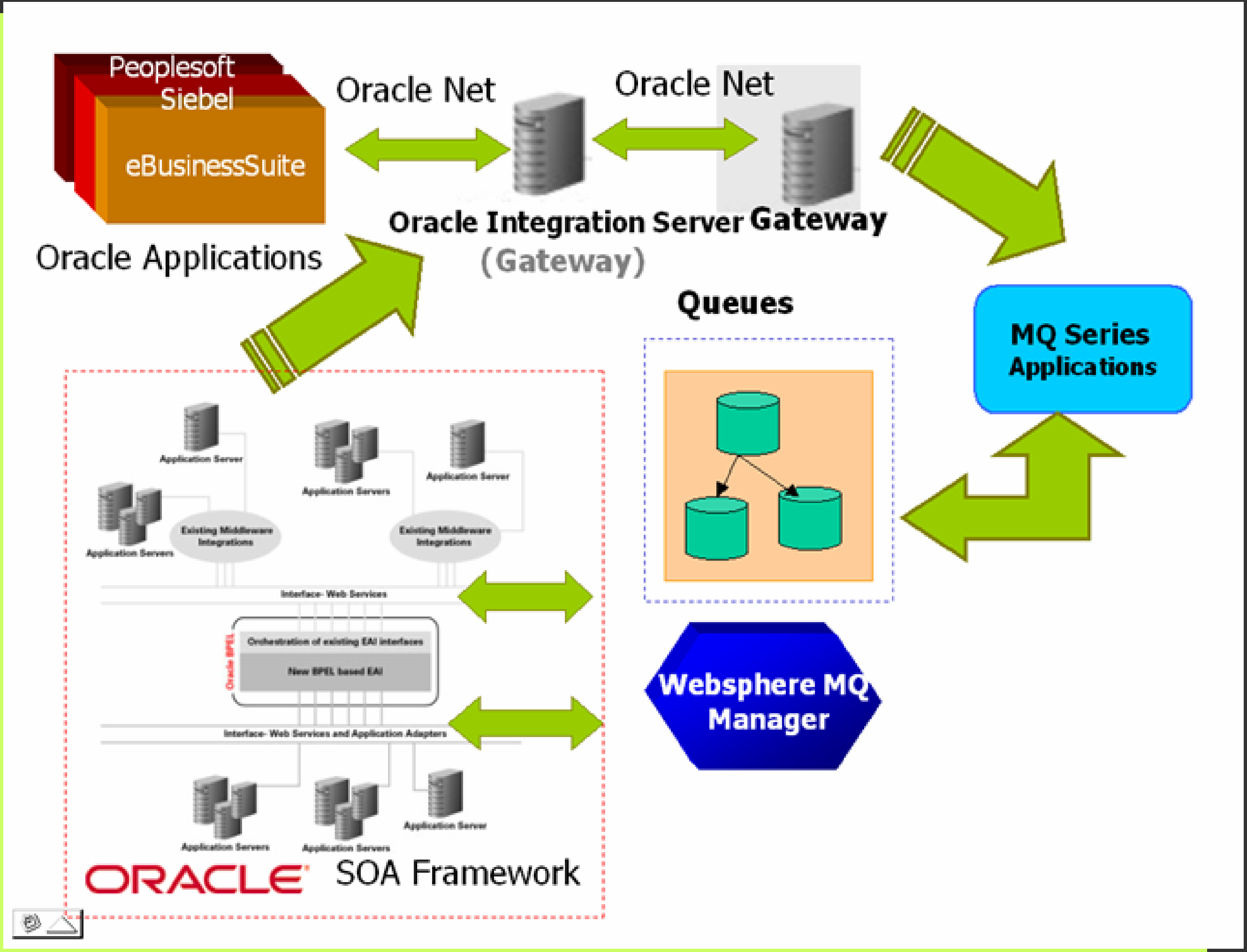
Business Models Concepts

- Message Queuing
- Data Replication (Logical Change Record)
- Data Protection
- Data Warehouse Loading
- Event Management and Notification
- Workflow
- Serializable Distributed Processing

Business Concepts

- Information Integration
- Automation and Business Event Management
- Message Queuing in SCM, ERP, CRM
- Data Protection and Information Hiding

Queuing Infrastructure



Types of Oracle Queues

- Based on Producer/Consumer Cardinality
  - Peer-to-Peer (P2P) Mode
  - Publish/Subscribe Mode
- Based on Persistency
  - Persistent
  - Non-Persistent

Types of Oracle Queues

- Based on Enqueue/Dequeue Capabilities
  - Normal
  - Exception

Types of Oracle Queues

- Based on Payload Data Type
  - ANYDATA
  - RAW
  - LOB
  - XML

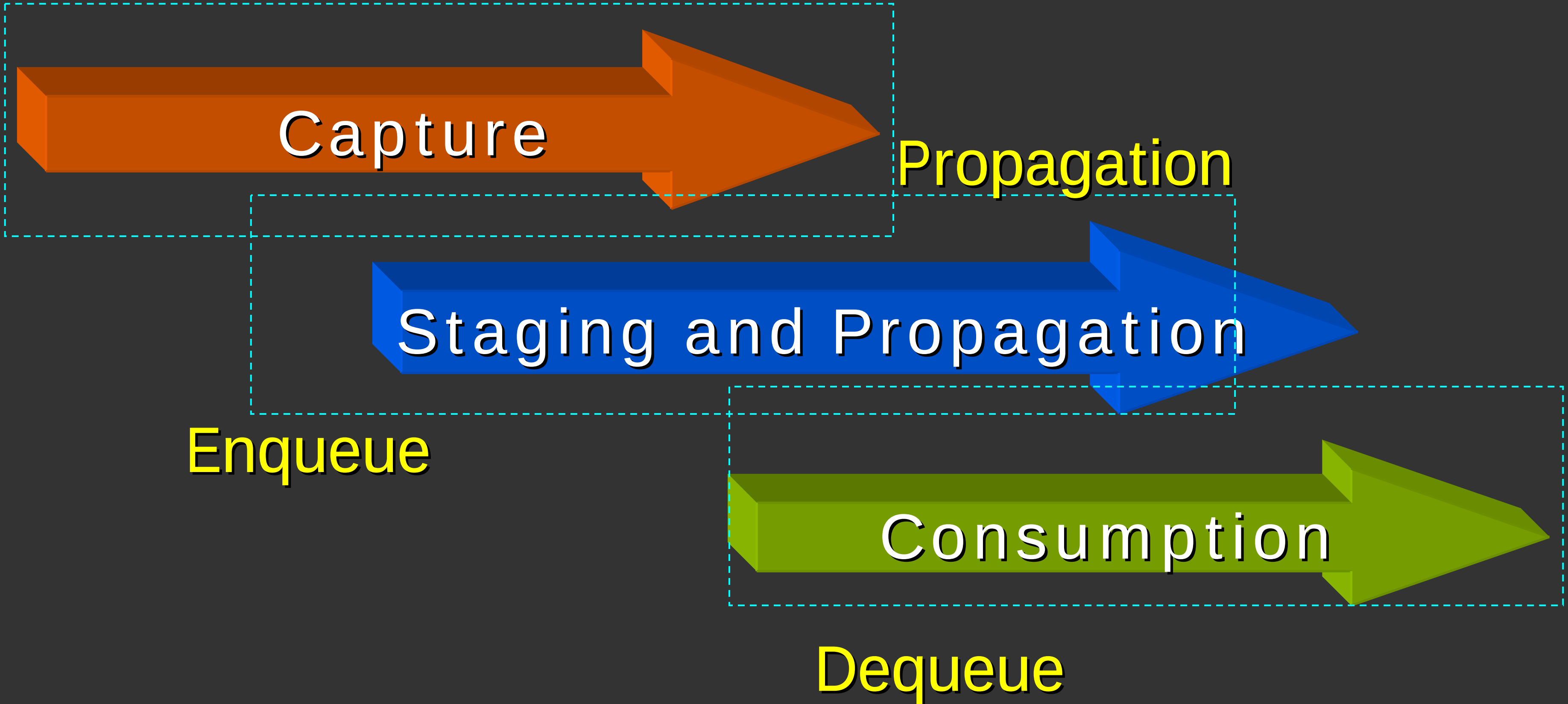
- On Based Transaction Type
  - Transactional
  - Non-Transactional

Oracle Streams AQ

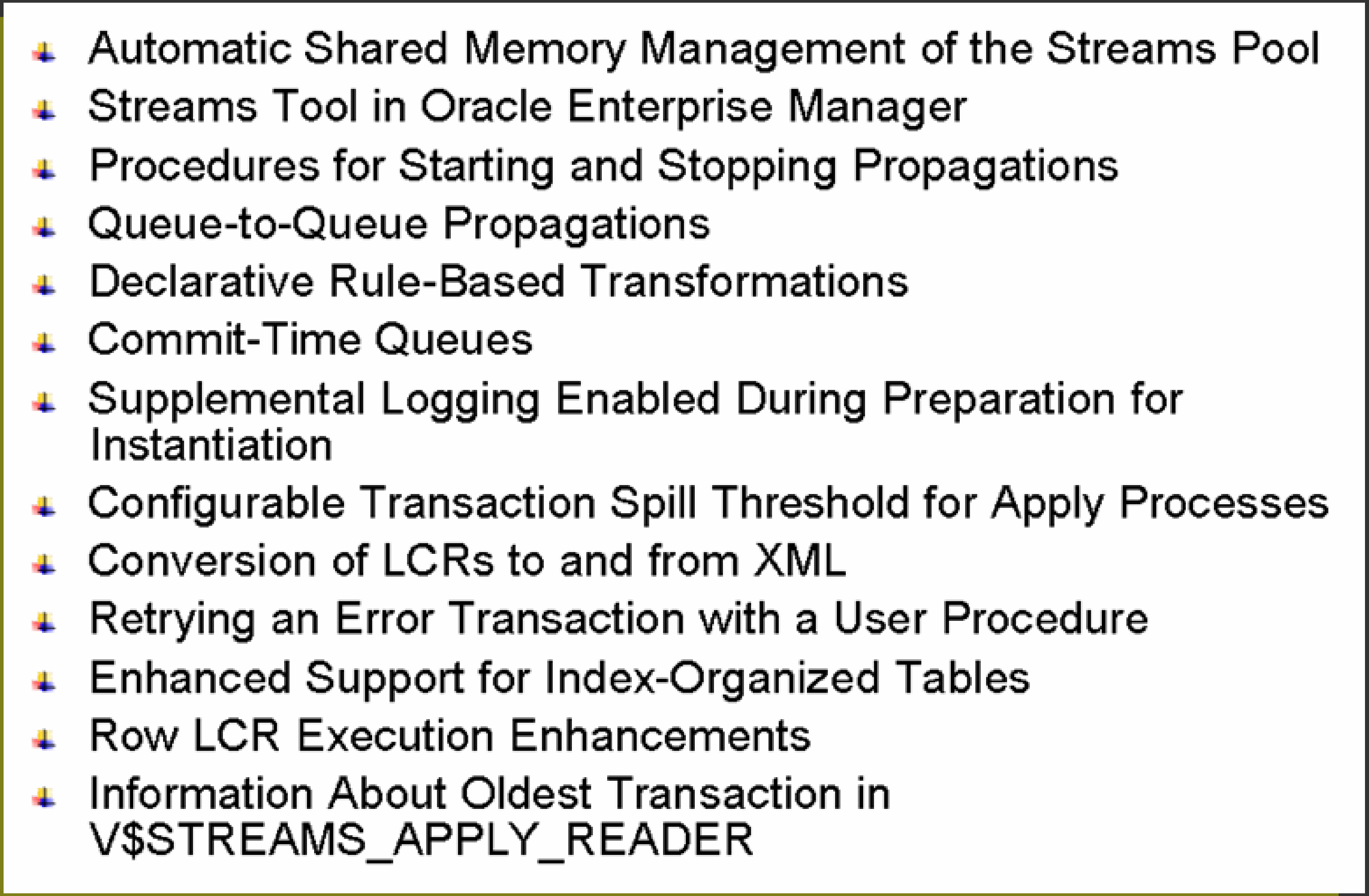
- "An application can enqueue messages that represent events into a queue explicitly, or a Streams capture process can capture database events and encapsulate them into messages called LCRs. These captured messages can be the results of DML or DDL changes. Propagations can propagate messages in a stream through multiple queues. Finally, a user application can dequeue messages explicitly, or a Streams apply process can dequeue messages implicitly. An apply process can reenqueue these messages explicitly into the same queue or a different queue if necessary. "

Oracle® Streams Concepts and Administration 10*g* Release 2 (10.2), Part Number B14229-02

AQ Downstream Model



Streams AQ Capabilities

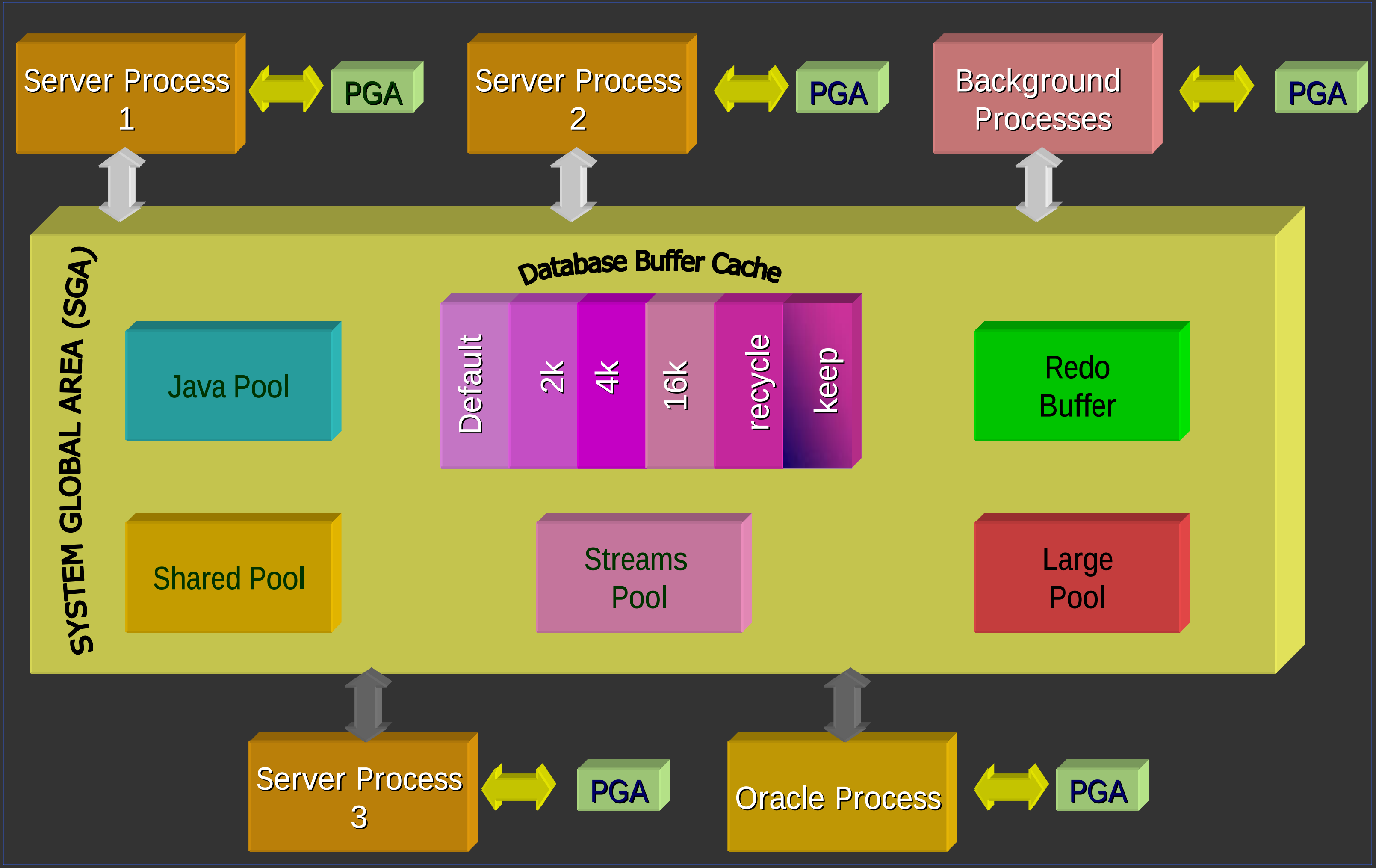


- Automatic Shared Memory Management of the Streams Pool
- Streams Tool in Oracle Enterprise Manager
- Procedures for Starting and Stopping Propagations
- Queue-to-Queue Propagations
- Declarative Rule-Based Transformations
- Commit-Time Queues
- Supplemental Logging Enabled During Preparation for Instantiation
- Configurable Transaction Spill Threshold for Apply Processes
- Conversion of LCRs to and from XML
- Retrying an Error Transaction with a User Procedure
- Enhanced Support for Index-Organized Tables
- Row LCR Execution Enhancements
- Information About Oldest Transaction in V$STREAMS_APPLY_READER

Architectural Considerations

- Oracle integration server
- Agent
- Queue table
- Queuing processes
- Listener configuration
- Database links
- Message-Oriented Middleware (MOM)

Architectural Considerations



Security Framework

- Rule-based Security
  - Object Level
  - Schema
  - Global
- Virtual Private Support

Security Framework

- Enhancements:
  - Database Volt
  - LDAP Support
  - XA Support
- Encryption Support via asymmetric authentication (PKI)

Planning the AQ Environment

- Oracle-based only or third-party, e.g., gateway-based or heterogeneous system involvement
- Transactional or non-transactional queue
- Peer-to-peer or Publish-Subscribe Mode (Broadcasting or Multicasting, i.e., custom recipient)

Planning the AQ Environment

- Propagation-type (queue-to-queue or queue-to-database link)
- Payload format
- Sending Mechanism (Producer Application)
- Receiving Mechanism (Consumer Application)

Planning the AQ Environment

- Model View Controller
  - Data Source, domain, data model
  - Presentation
  - Controller / Mediator
- Payload content management for intelligent rule-based filtering or routing

Planning the AQ Environment

- Queue browsing without consumption
- Queue consumption and removal
- Queue consumption without payload removal for auditing, non-repudiation, or logging.

System Requirements

- Configuring:
  - PATH, LD_LIBRARY_PATH or equivalent platform parameters, as needed
  - Streams pool instance and/or aq_tm_processes (9i only) > 0.

System Requirements

- Configuring:
  - Extproc listener
  - Messaging Gateway, if applicable, involves gateway software installation and packages, including messaging home.

System Requirements

- Configuring:
  - Create AQ user and administrator with appropriate privileges, namely, AQ_USER_ROLE and AQ_ADMINISTRATOR_ROLE
  - Database links accordingly
  - Heterogeneous Services, if applicable (involved package and instance configuration)

Software Requirements

- Certified OS Platform
- Oracle Streams AQ
- Oracle Streams AQ Gateway
    - Procedural Gateway (Websphere MQ/Tibco)
    - Transparent Gateway (SQL Server)
- Heterogeneous Services Gateway
- Configure Gateway homes with API provided

Initialization Parameters

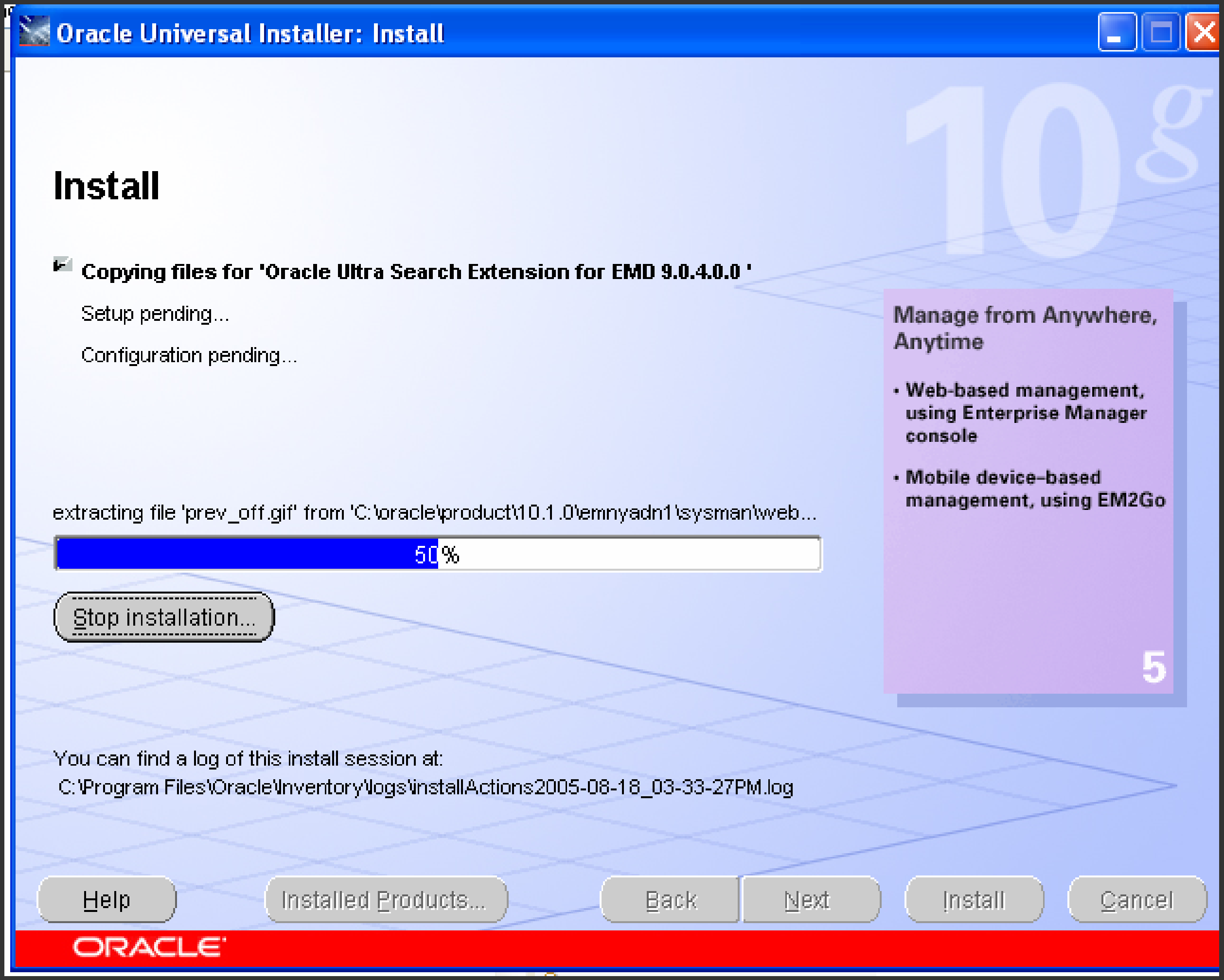
- Oracle9i
  - Aq_tm_processes=n, 1<= n <=10
  - Qmnc, master process
  - Qxxx, spawned slave processes
  - =0, then no queue monitoring

Initialization Parameters

- Oracle10g
  - Streams_pool_size, configured with dynamic memory management. Recommended default setting about 10% of shared_pool_size parameter.

Installation and Configuration

Information Integration Components



Oracle Universal Installer: Install

Install

Copying files for 'Oracle Ultra Search Extension for EMD 9.0.4.0.0 '

Setup pending...

Configuration pending...

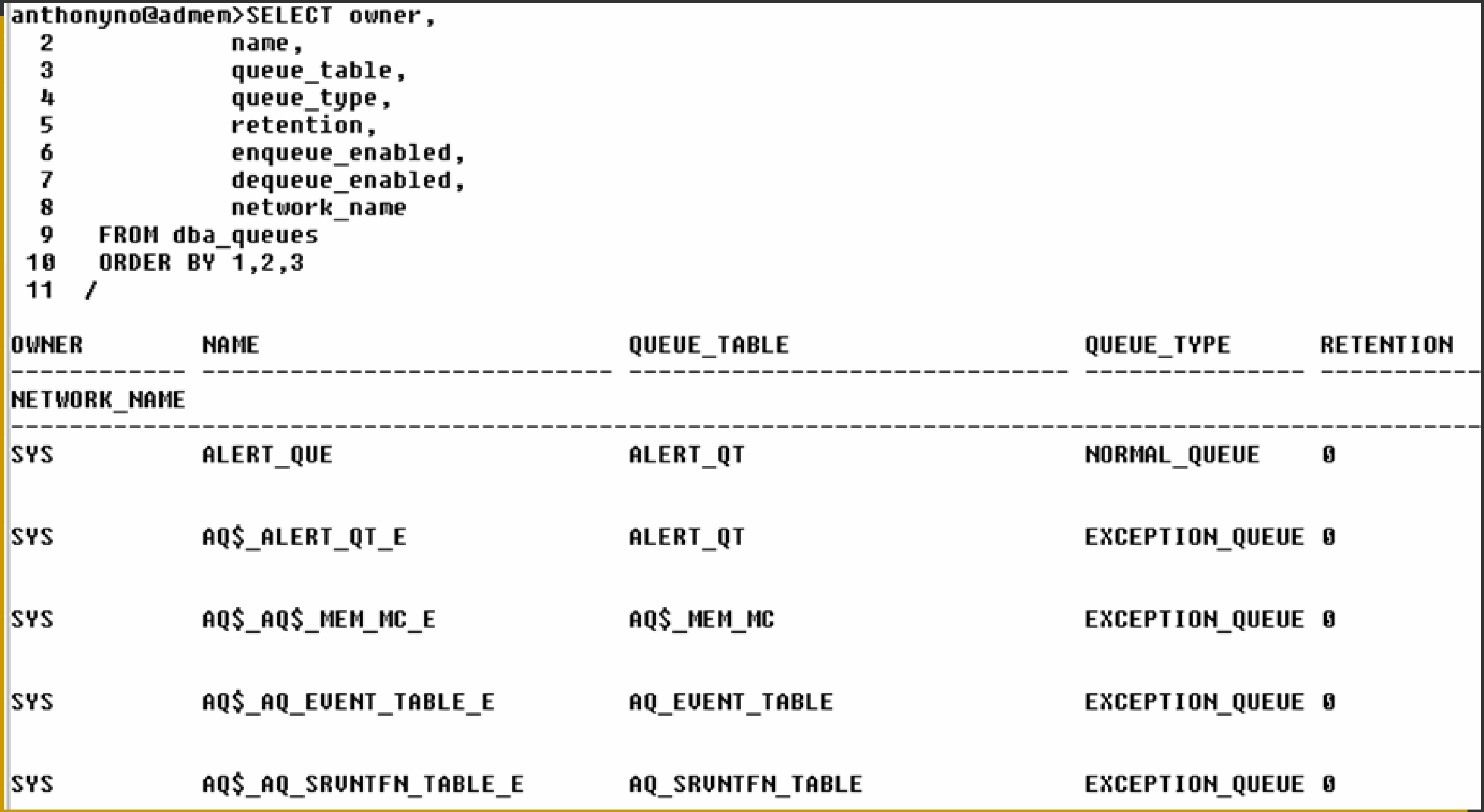
extracting file 'prev_off.gif' from 'C:\oracle\product\10.1.0\emnyadn1\sysman\web...

50%

You can find a log of this install session at:
C:\Program Files\Oracle\Inventory\logs\installActions2005-08-18_03-33-27PM.log

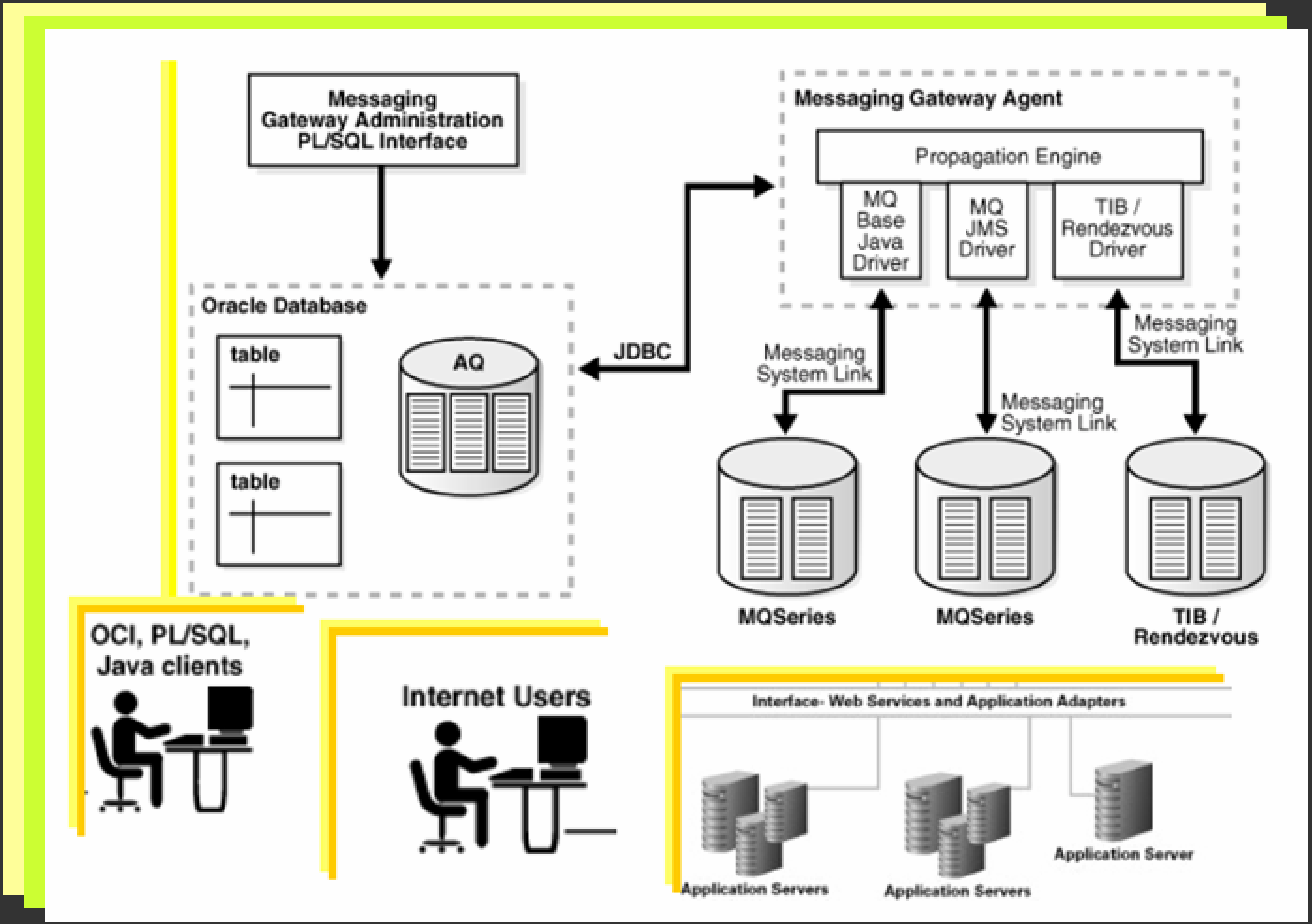
Stop installation...
10g
Manage from Anywhere, Anytime
• Web-based management, using Enterprise Manager console
• Mobile device-based management, using EM2Go
5
Help
Installed Products...
Back
Next
Install
Cancel
ORACLE

Data Dictionary Views

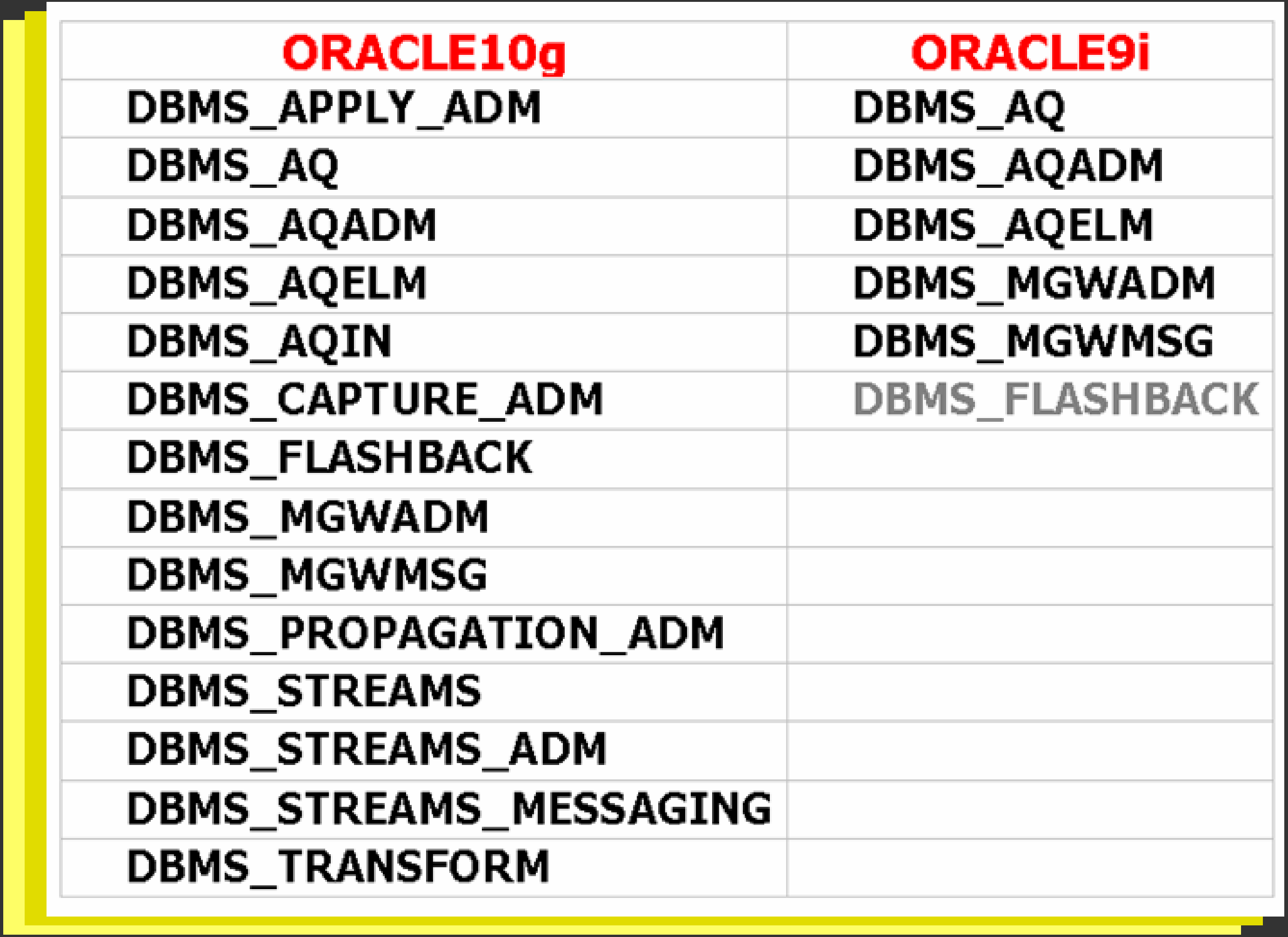


```
anthonyno@admem>SELECT owner,
  2             name,
  3             queue_table,
  4             queue_type,
  5             retention,
  6             enqueue_enabled,
  7             dequeue_enabled,
  8             network_name
  9    FROM dba_queues
 10    ORDER BY 1,2,3
 11   /

OWNER        NAME                          QUEUE_TABLE                    QUEUE_TYPE      RETENTION
------------ ----------------------------- ------------------------------ --------------- ----------
NETWORK_NAME
----------------------------------------------------------------------------------------------------
SYS          ALERT_QUE                     ALERT_QT                       NORMAL_QUEUE    0


SYS          AQ$_ALERT_QT_E                ALERT_QT                       EXCEPTION_QUEUE 0


SYS          AQ$_AQ$_MEM_MC_E              AQ$_MEM_MC                     EXCEPTION_QUEUE 0


SYS          AQ$_AQ_EVENT_TABLE_E          AQ_EVENT_TABLE                 EXCEPTION_QUEUE 0


SYS          AQ$_AQ_SRVNTFN_TABLE_E        AQ_SRVNTFN_TABLE               EXCEPTION_QUEUE 0
```

Heterogeneous Productivity

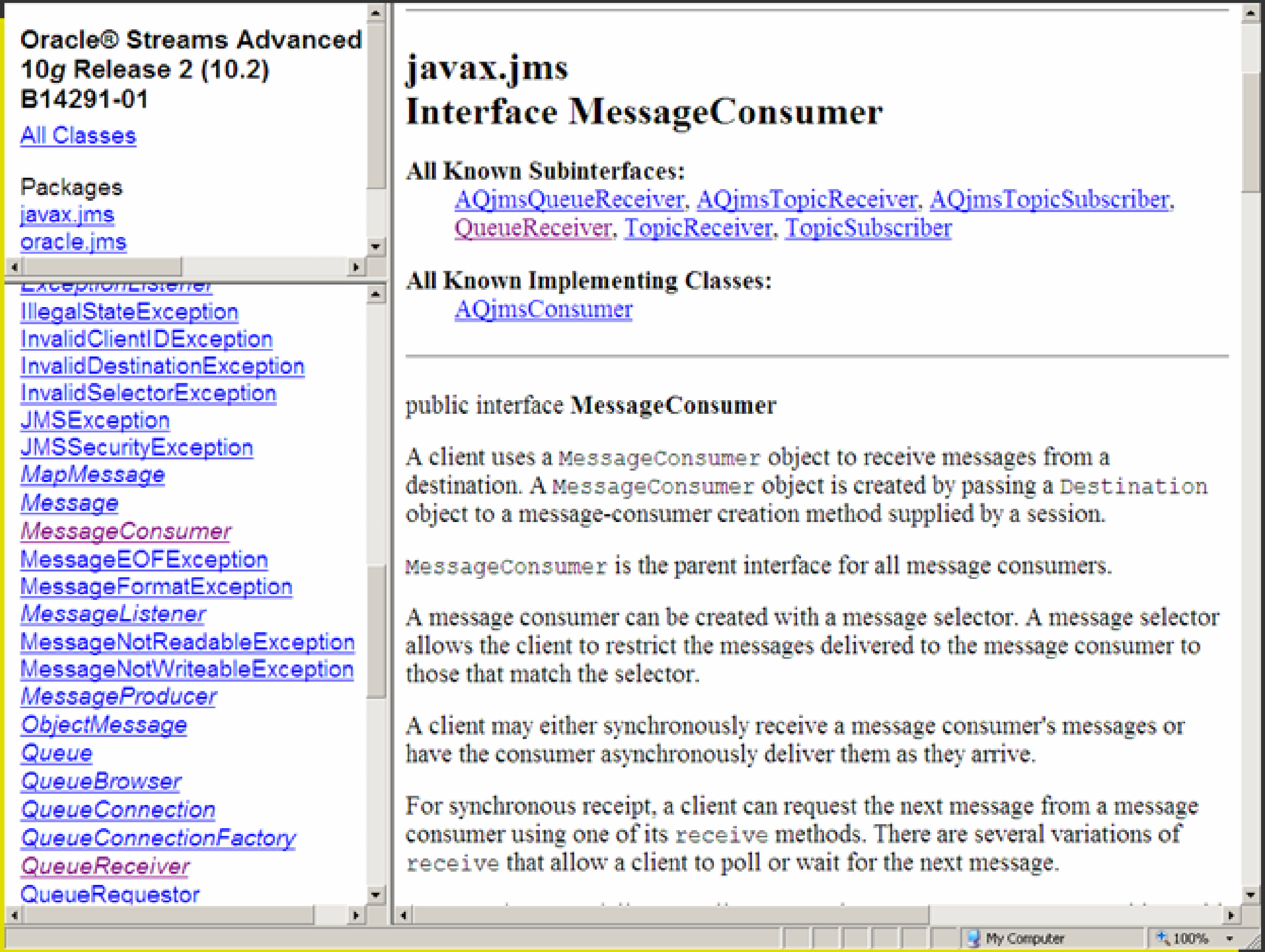


PL/SQL Supplied Packages



ORACLE10g	ORACLE9i
DBMS_APPLY_ADM	DBMS_AQ
DBMS_AQ	DBMS_AQADM
DBMS_AQADM	DBMS_AQELM
DBMS_AQELM	DBMS_MGWADM
DBMS_AQIN	DBMS_MGWMSG
DBMS_CAPTURE_ADM	DBMS_FLASHBACK
DBMS_FLASHBACK	
DBMS_MGWADM	
DBMS_MGWMSG	
DBMS_PROPAGATION_ADM	
DBMS_STREAMS	
DBMS_STREAMS_ADM	
DBMS_STREAMS_MESSAGING	
DBMS_TRANSFORM	

Java Supplied Packages



Oracle® Streams Advanced
10*g* Release 2 (10.2)
B14291-01

All Classes

Packages
javax.jms
oracle.jms

ExceptionListener
IllegalStateException
InvalidClientIDException
InvalidDestinationException
InvalidSelectorException
JMSException
JMSSecurityException
MapMessage
Message
MessageConsumer
MessageEOFException
MessageFormatException
MessageListener
MessageNotReadableException
MessageNotWriteableException
MessageProducer
ObjectMessage
Queue
QueueBrowser
QueueConnection
QueueConnectionFactory
QueueReceiver
QueueRequestor

javax.jms
Interface MessageConsumer

All Known Subinterfaces:
AQjmsQueueReceiver, AQjmsTopicReceiver, AQjmsTopicSubscriber, QueueReceiver, TopicReceiver, TopicSubscriber

All Known Implementing Classes:
AQjmsConsumer

public interface **MessageConsumer**

A client uses a `MessageConsumer` object to receive messages from a destination. A `MessageConsumer` object is created by passing a `Destination` object to a message-consumer creation method supplied by a session.

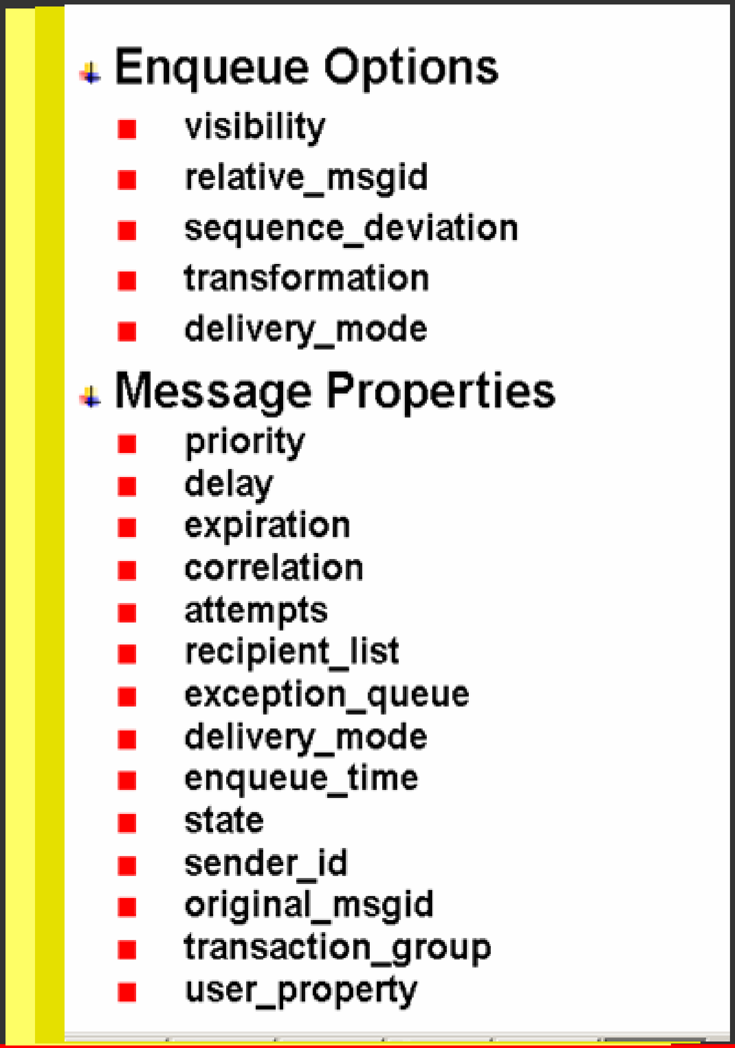
`MessageConsumer` is the parent interface for all message consumers.

A message consumer can be created with a message selector. A message selector allows the client to restrict the messages delivered to the message consumer to those that match the selector.

A client may either synchronously receive a message consumer's messages or have the consumer asynchronously deliver them as they arrive.

For synchronous receipt, a client can request the next message from a message consumer using one of its `receive` methods. There are several variations of `receive` that allow a client to poll or wait for the next message.

Enqueuing Concepts

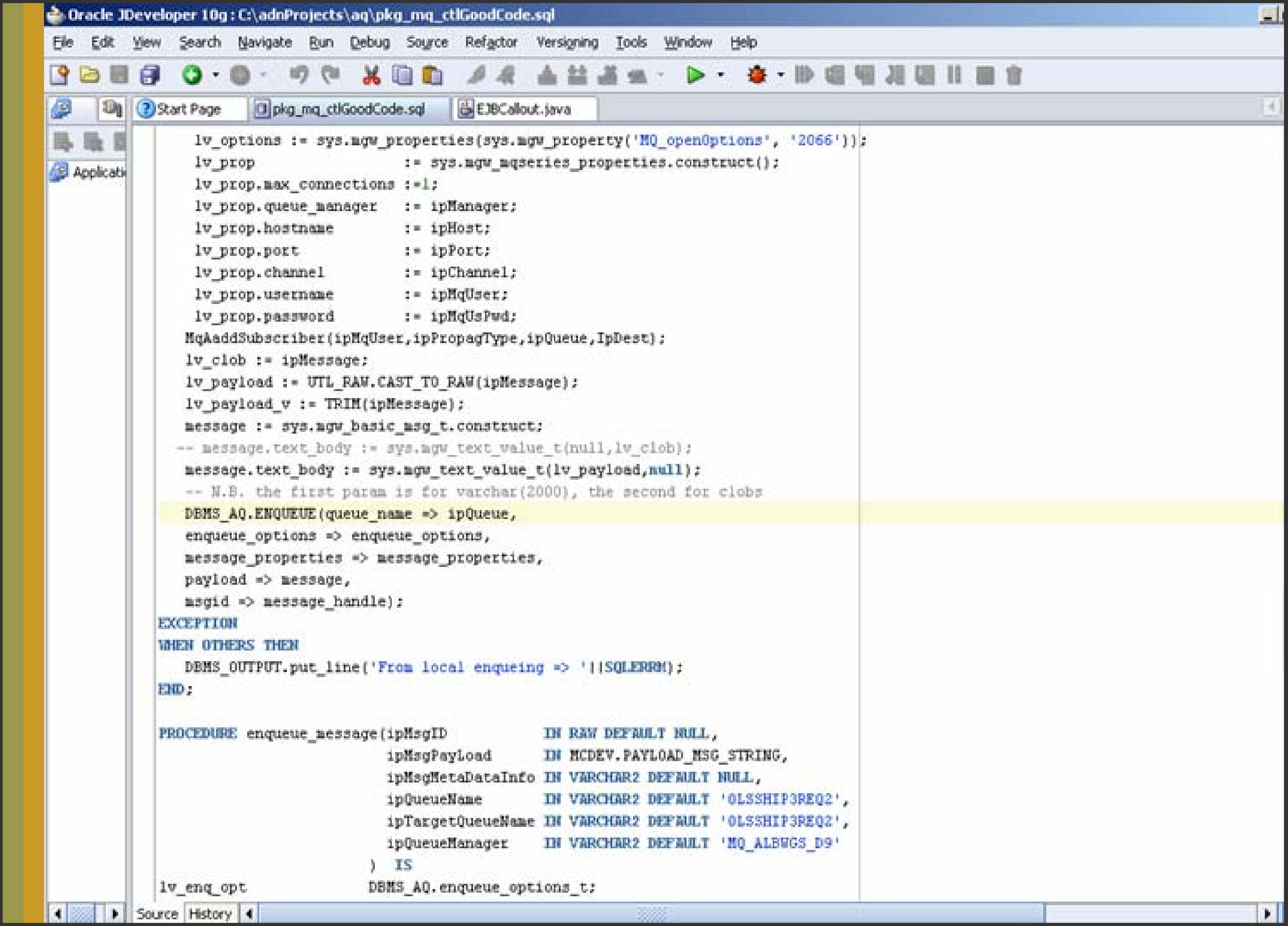


- Enqueue Options
  - visibility
  - relative_msgid
  - sequence_deviation
  - transformation
  - delivery_mode
- Message Properties
  - priority
  - delay
  - expiration
  - correlation
  - attempts
  - recipient_list
  - exception_queue
  - delivery_mode
  - enqueue_time
  - state
  - sender_id
  - original_msgid
  - transaction_group
  - user_property

Monitoring Staging

- Message delay
- Message expiration
- Retry delay
- Garbage collection for the queue table
- Retention and Message History
- Cleaning Up Message Queues
- Tracking and Event Journals
- Non-repudiation
- Queue Forwarding

Programming AQ

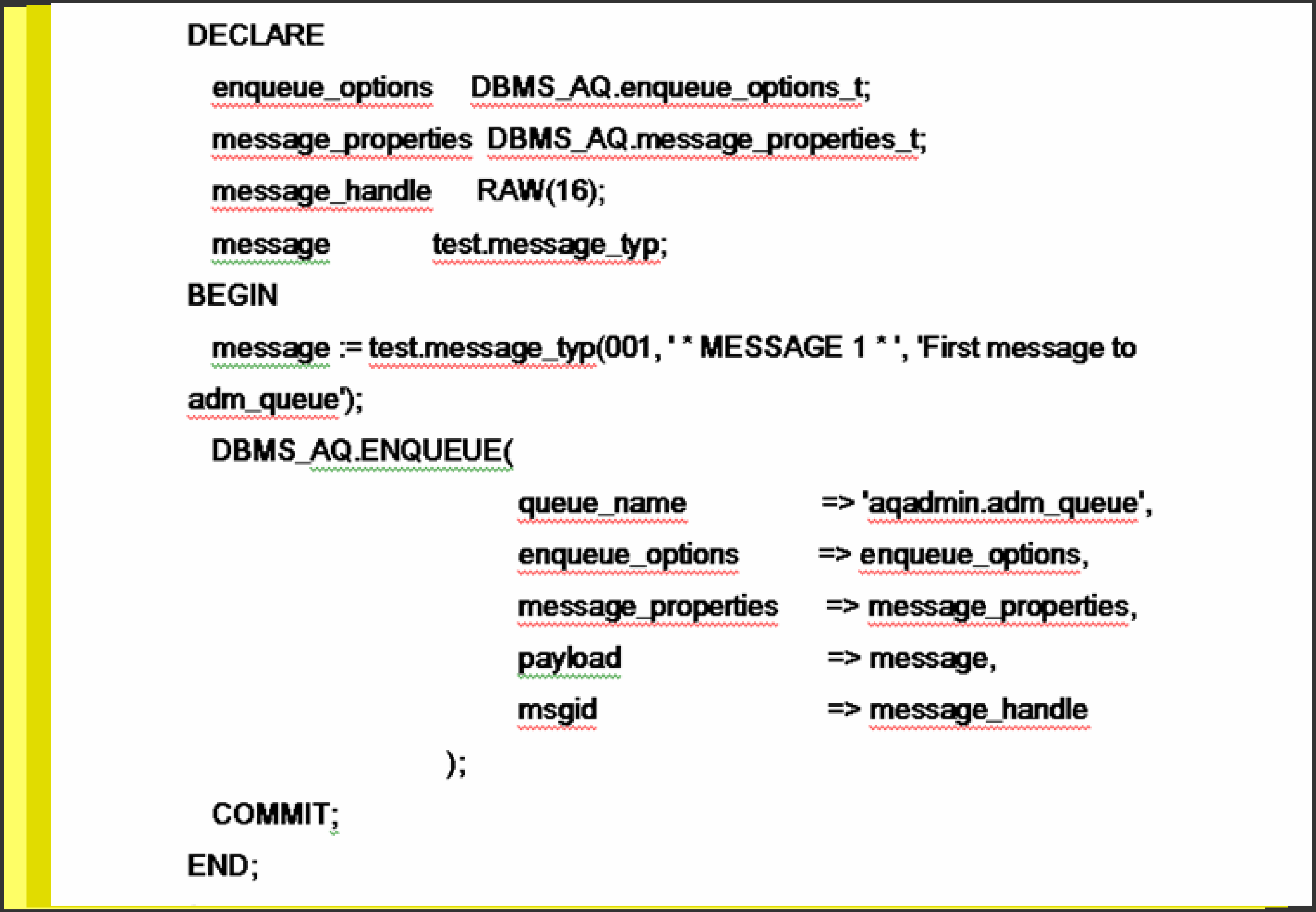

Oracle JDeveloper 10g : C:\adnProjects\aq\pkg_mq_ctlGoodCode.sql
File Edit View Search Navigate Run Debug Source Refactor Versioning Tools Window Help
Start Page
pkg_mq_ctlGoodCode.sql
EJBCallout.java
Applicati
Source
History

```
    lv_options := sys.mgw_properties(sys.mgw_property('MQ_openOptions', '2066'));
    lv_prop                 := sys.mgw_mqseries_properties.construct();
    lv_prop.max_connections :=1;
    lv_prop.queue_manager   := ipManager;
    lv_prop.hostname        := ipHost;
    lv_prop.port            := ipPort;
    lv_prop.channel         := ipChannel;
    lv_prop.username        := ipMqUser;
    lv_prop.password        := ipMqUsPwd;
   MqAaddSubscriber(ipMqUser,ipPropagType,ipQueue,IpDest);
   lv_clob := ipMessage;
   lv_payload := UTL_RAW.CAST_TO_RAW(ipMessage);
   lv_payload_v := TRIM(ipMessage);
   message := sys.mgw_basic_msg_t.construct;
  -- message.text_body := sys.mgw_text_value_t(null,lv_clob);
   message.text_body := sys.mgw_text_value_t(lv_payload,null);
   -- N.B. the first param is for varchar(2000), the second for clobs
   DBMS_AQ.ENQUEUE(queue_name => ipQueue,
   enqueue_options => enqueue_options,
   message_properties => message_properties,
   payload => message,
   msgid => message_handle);
EXCEPTION
WHEN OTHERS THEN
   DBMS_OUTPUT.put_line('From local enqueing => '||SQLERRM);
END;

PROCEDURE enqueue_message(ipMsgID            IN RAW DEFAULT NULL,
                          ipMsgPayload       IN MCDEV.PAYLOAD_MSG_STRING,
                          ipMsgMetaDataInfo  IN VARCHAR2 DEFAULT NULL,
                          ipQueueName        IN VARCHAR2 DEFAULT 'OLSSHIP3REQ2',
                          ipTargetQueueName  IN VARCHAR2 DEFAULT 'OLSSHIP3REQ2',
                          ipQueueManager     IN VARCHAR2 DEFAULT 'MQ_ALBWGS_D9'
                        )  IS
lv_enq_opt              DBMS_AQ.enqueue_options_t;
```

1

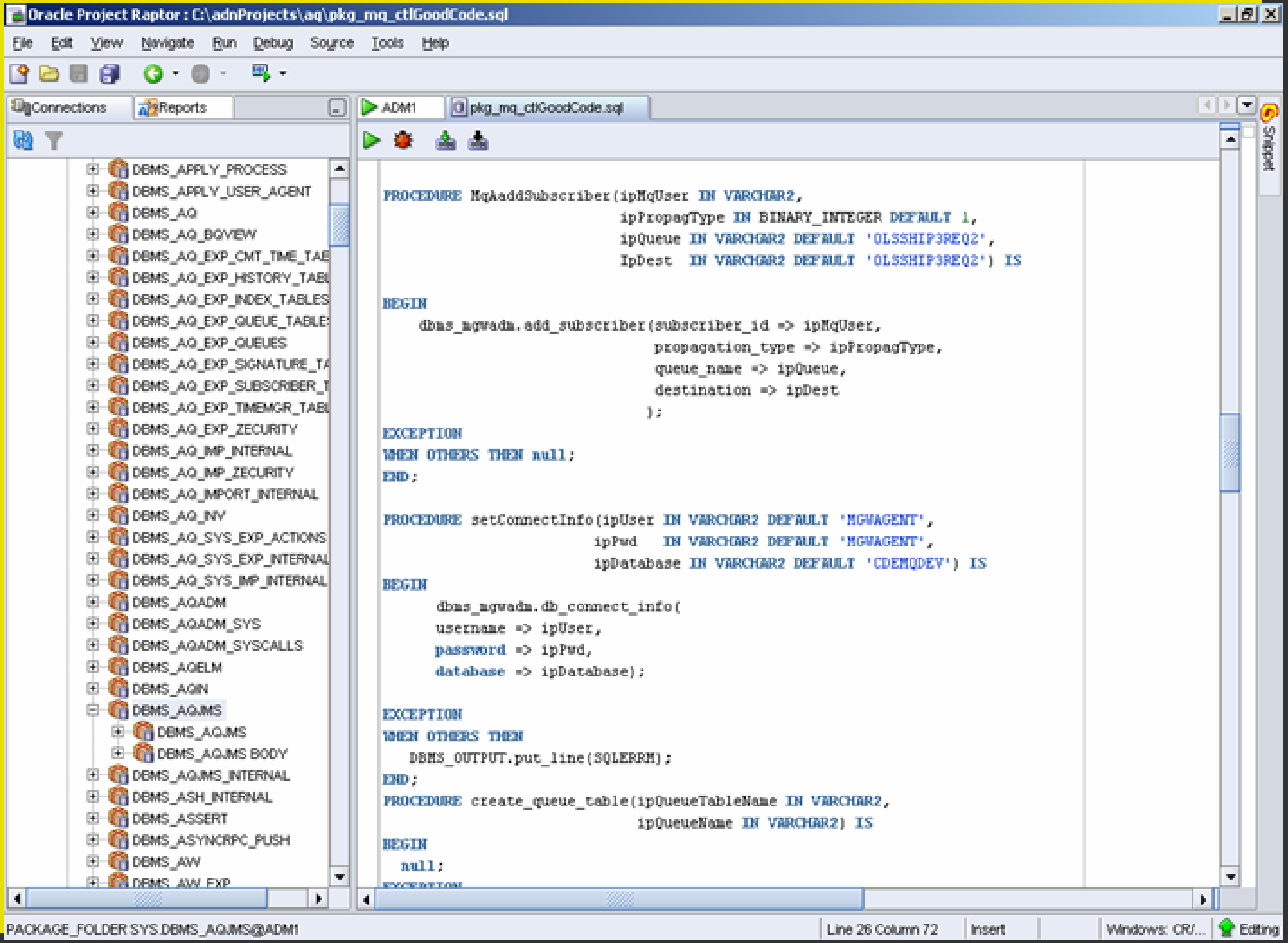
Programming AQ

2



```
DECLARE
   enqueue_options      DBMS_AQ.enqueue_options_t;
   message_properties DBMS_AQ.message_properties_t;
   message_handle     RAW(16);
   message             test.message_typ;
BEGIN
   message := test.message_typ(001, ' * MESSAGE 1 * ', 'First message to
adm_queue');
   DBMS_AQ.ENQUEUE(
                                   queue_name              => 'aqadmin.adm_queue',
                                   enqueue_options        => enqueue_options,
                                   message_properties     => message_properties,
                                   payload                 => message,
                                   msgid                   => message_handle
                          );
   COMMIT;
END;
```

Programming AQ


Oracle Project Raptor : C:\adnProjects\aq\pkg_mq_ctlGoodCode.sql
File Edit View Navigate Run Debug Source Tools Help
Connections
Reports
ADM1
pkg_mq_ctlGoodCode.sql
Snippet

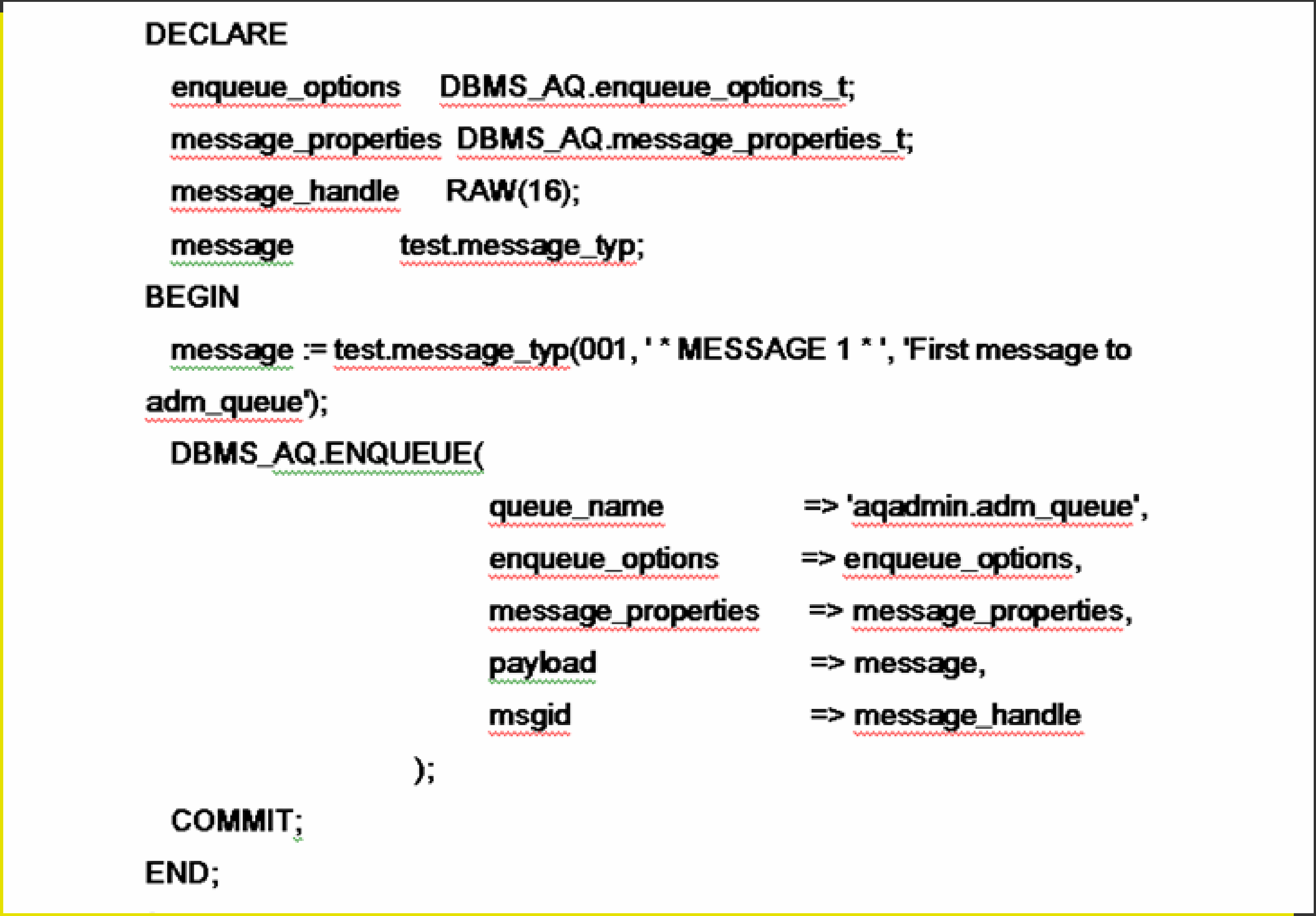
DBMS_APPLY_PROCESS
DBMS_APPLY_USER_AGENT
DBMS_AQ
DBMS_AQ_BQVIEW
DBMS_AQ_EXP_CMT_TIME_TAE
DBMS_AQ_EXP_HISTORY_TABL
DBMS_AQ_EXP_INDEX_TABLES
DBMS_AQ_EXP_QUEUE_TABLE
DBMS_AQ_EXP_QUEUES
DBMS_AQ_EXP_SIGNATURE_TA
DBMS_AQ_EXP_SUBSCRIBER_T
DBMS_AQ_EXP_TIMEMGR_TABL
DBMS_AQ_EXP_ZECURITY
DBMS_AQ_IMP_INTERNAL
DBMS_AQ_IMP_ZECURITY
DBMS_AQ_IMPORT_INTERNAL
DBMS_AQ_INV
DBMS_AQ_SYS_EXP_ACTIONS
DBMS_AQ_SYS_EXP_INTERNAL
DBMS_AQ_SYS_IMP_INTERNAL
DBMS_AQADM
DBMS_AQADM_SYS
DBMS_AQADM_SYSCALLS
DBMS_AQELM
DBMS_AQIN
DBMS_AQJMS
  DBMS_AQJMS
  DBMS_AQJMS BODY
DBMS_AQJMS_INTERNAL
DBMS_ASH_INTERNAL
DBMS_ASSERT
DBMS_ASYNCRPC_PUSH
DBMS_AW
DBMS_AW_EXP

```
PROCEDURE MqAaddSubscriber(ipMqUser IN VARCHAR2,
                           ipPropagType IN BINARY_INTEGER DEFAULT 1,
                           ipQueue IN VARCHAR2 DEFAULT 'OLSSHIP3REQ2',
                           IpDest  IN VARCHAR2 DEFAULT 'OLSSHIP3REQ2') IS

BEGIN
    dbms_mgwadm.add_subscriber(subscriber_id => ipMqUser,
                               propagation_type => ipPropagType,
                               queue_name => ipQueue,
                               destination => ipDest
                              );
EXCEPTION
WHEN OTHERS THEN null;
END;

PROCEDURE setConnectInfo(ipUser IN VARCHAR2 DEFAULT 'MGWAGENT',
                         ipPwd   IN VARCHAR2 DEFAULT 'MGWAGENT',
                         ipDatabase IN VARCHAR2 DEFAULT 'CDEMQDEV') IS
BEGIN
      dbms_mgwadm.db_connect_info(
      username => ipUser,
      password => ipPwd,
      database => ipDatabase);

EXCEPTION
WHEN OTHERS THEN
   DBMS_OUTPUT.put_line(SQLERRM);
END;
PROCEDURE create_queue_table(ipQueueTableName IN VARCHAR2,
                             ipQueueName IN VARCHAR2) IS
BEGIN
  null;
```

PACKAGE_FOLDER SYS.DBMS_AQJMS@ADM1 | Line 26 Column 72 | Insert | Windows: CR/... | Editing

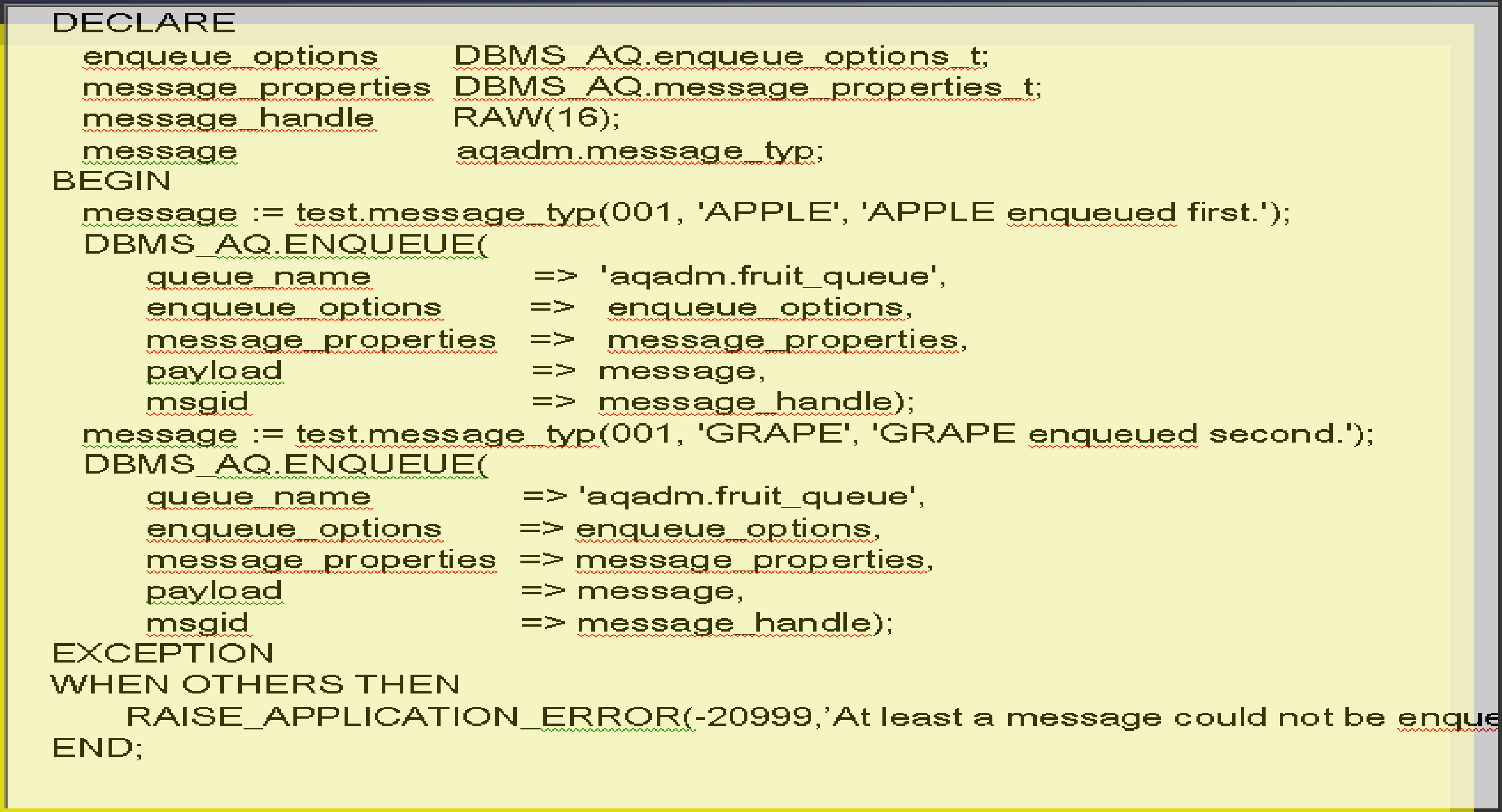
3

Programming AQ



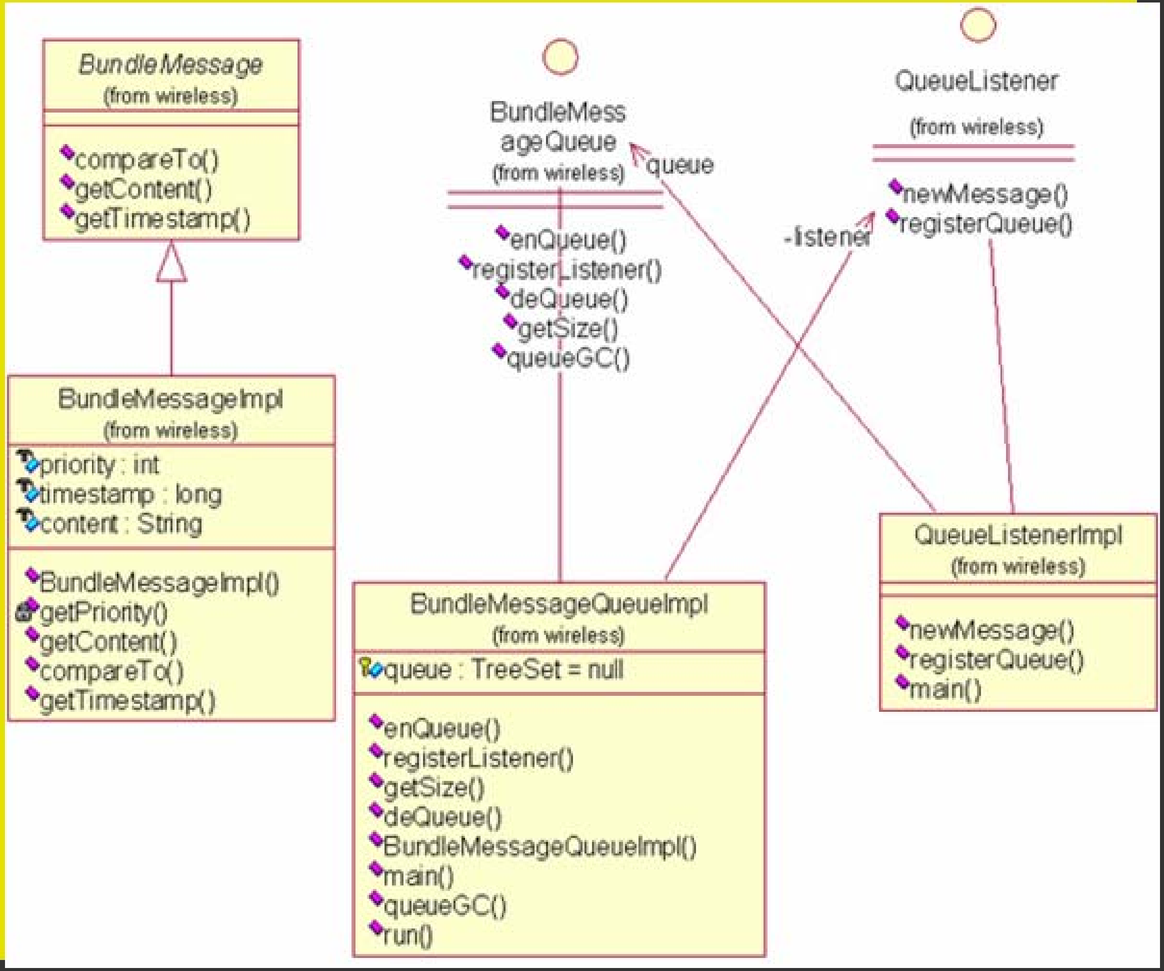
```
DECLARE
  enqueue_options    DBMS_AQ.enqueue_options_t;
  message_properties DBMS_AQ.message_properties_t;
  message_handle     RAW(16);
  message            test.message_typ;
BEGIN
  message := test.message_typ(001, ' * MESSAGE 1 * ', 'First message to
adm_queue');
  DBMS_AQ.ENQUEUE(
                queue_name          => 'aqadmin.adm_queue',
                enqueue_options     => enqueue_options,
                message_properties  => message_properties,
                payload             => message,
                msgid               => message_handle
          );
  COMMIT;
END;
```

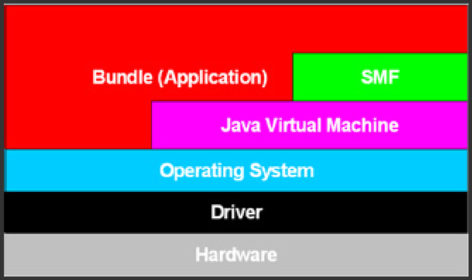
Using AQ



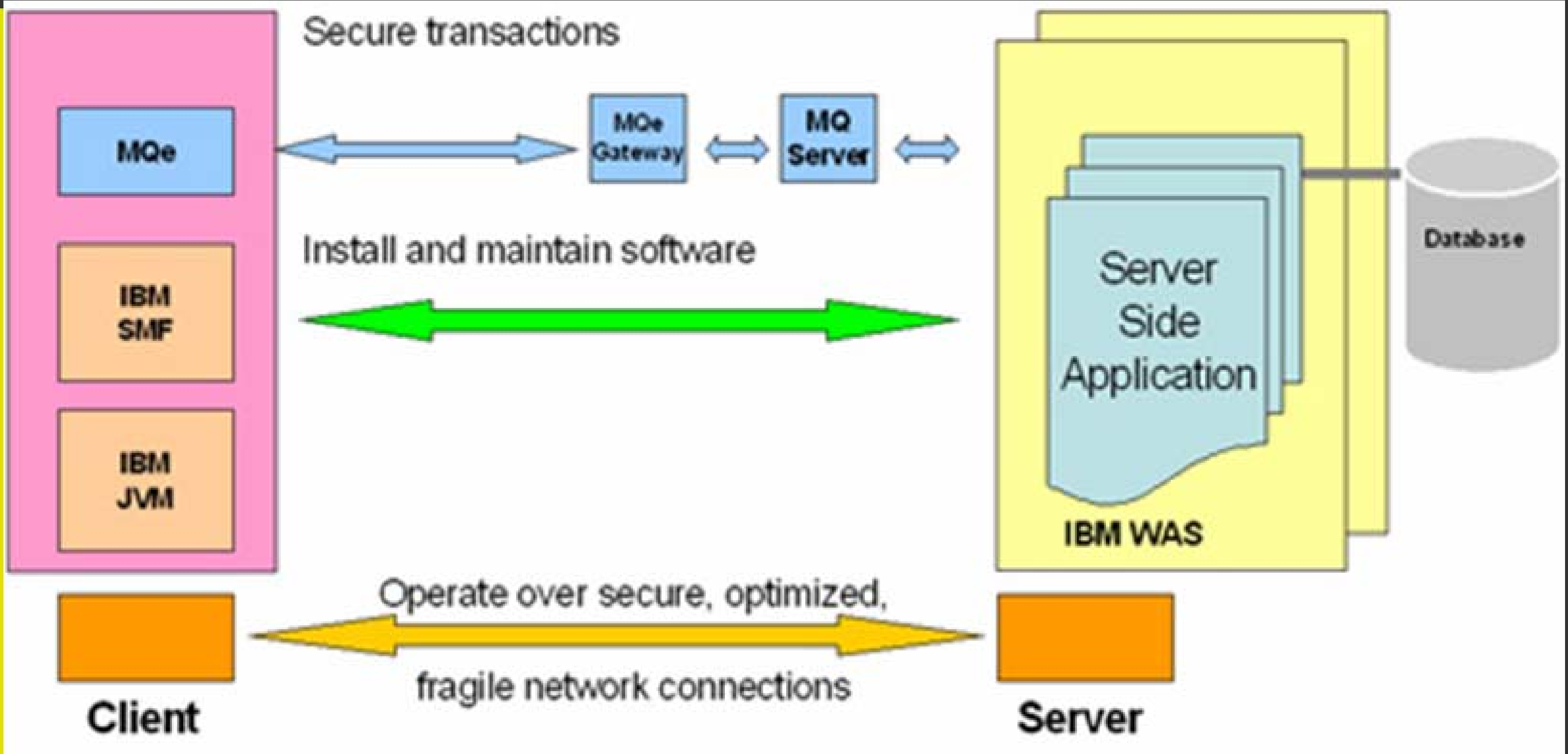
```
DECLARE
   enqueue_options       DBMS_AQ.enqueue_options_t;
   message_properties  DBMS_AQ.message_properties_t;
   message_handle        RAW(16);
   message                      aqadm.message_typ;
BEGIN
   message := test.message_typ(001, 'APPLE', 'APPLE enqueued first.');
   DBMS_AQ.ENQUEUE(
         queue_name                 =>  'aqadm.fruit_queue',
         enqueue_options        =>   enqueue_options,
         message_properties  =>   message_properties,
         payload                        =>  message,
         msgid                           =>  message_handle);
   message := test.message_typ(001, 'GRAPE', 'GRAPE enqueued second.');
   DBMS_AQ.ENQUEUE(
         queue_name                =>  'aqadm.fruit_queue',
         enqueue_options       =>  enqueue_options,
         message_properties =>  message_properties,
         payload                       =>  message,
         msgid                          =>  message_handle);
EXCEPTION
WHEN OTHERS THEN
      RAISE_APPLICATION_ERROR(-20999,'At least a message could not be enque
END;
```

Using AQ with MQ



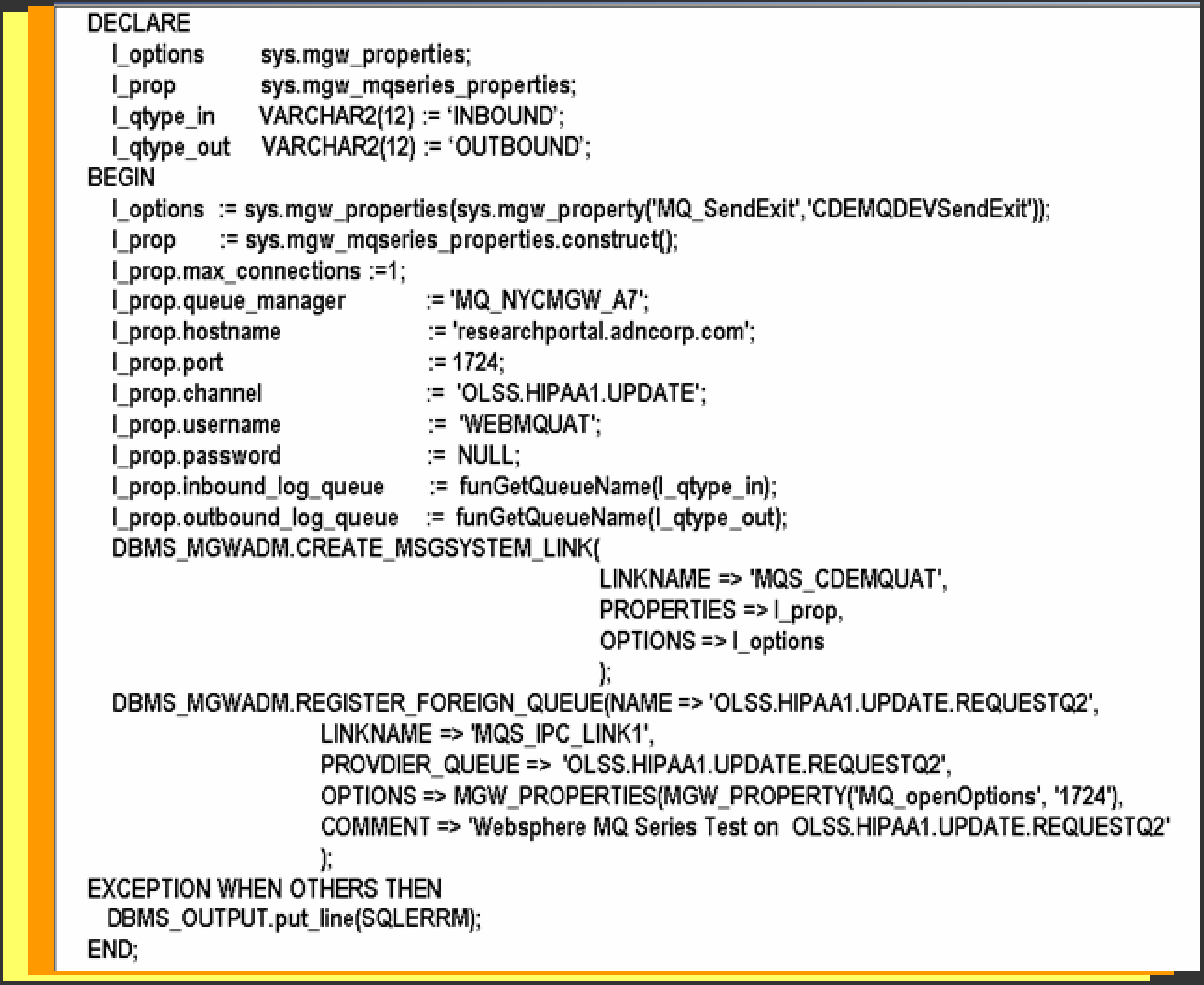


Using AQ with MQ



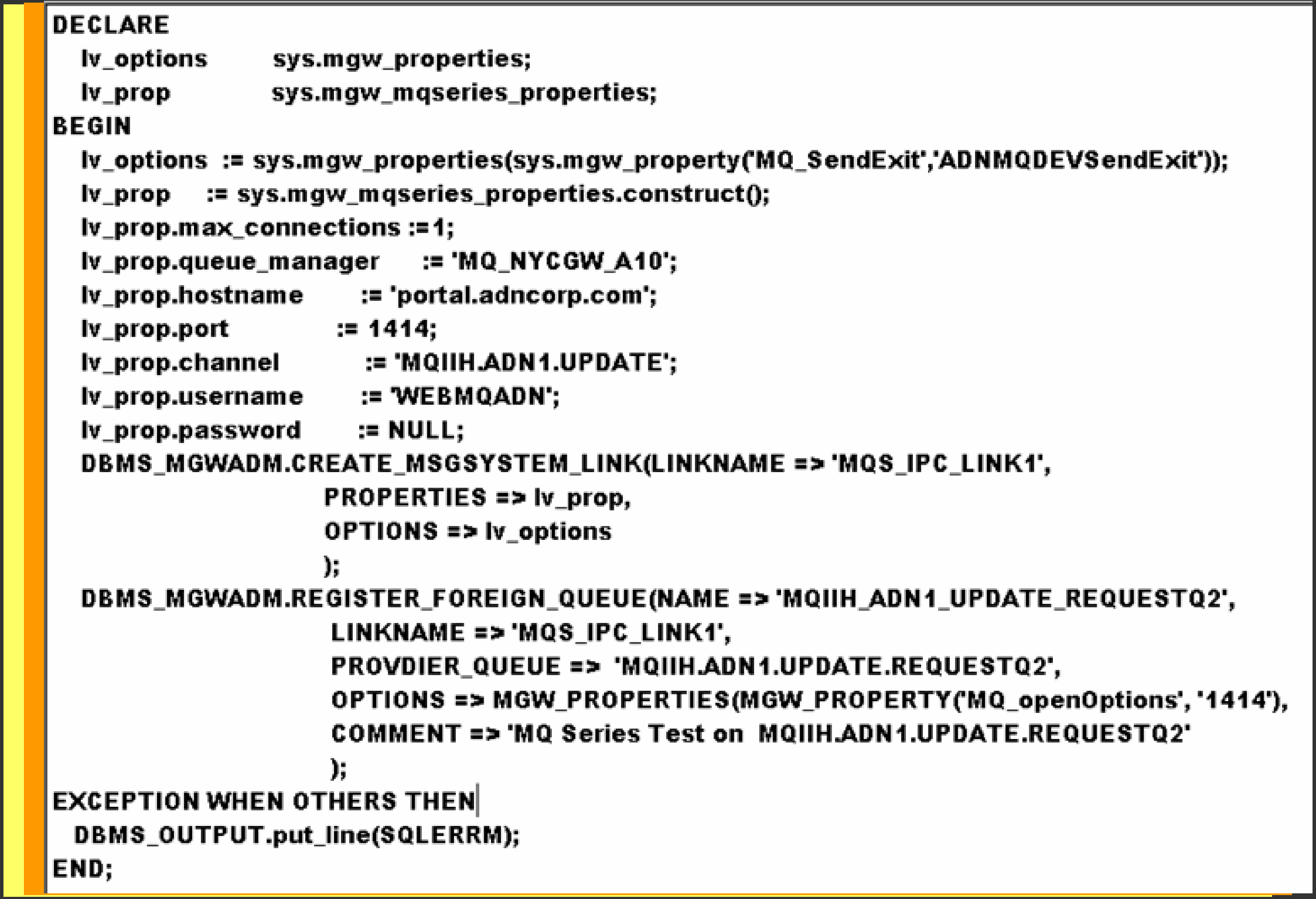
- IBM SMF: IBM Service Management Framework
- IBM JVM: IBM Java Virtual Machine
- IBM WAS: IBM WebSphere Application Server

Using AQ with MQ



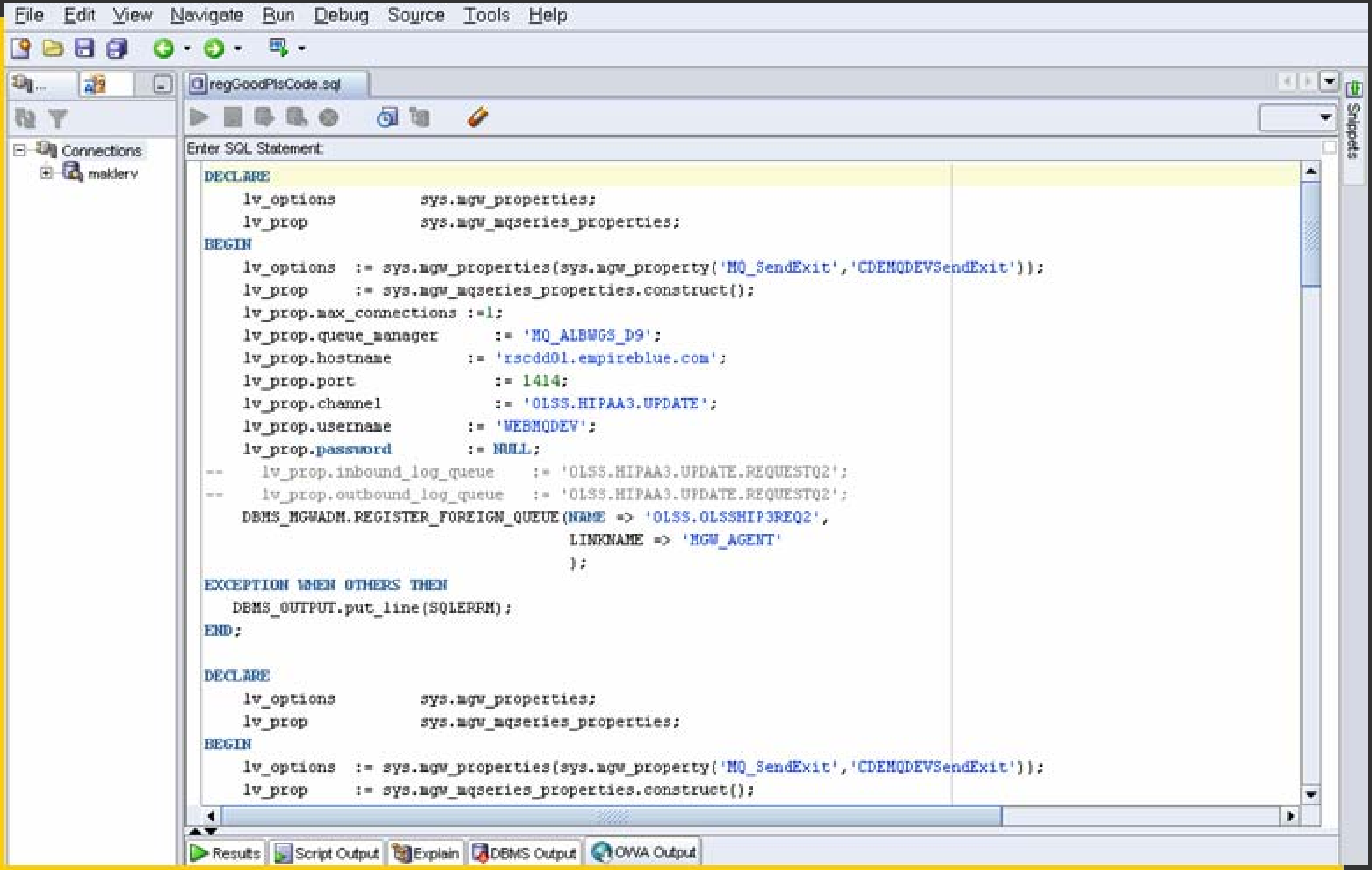
```
DECLARE
  l_options       sys.mgw_properties;
  l_prop          sys.mgw_mqseries_properties;
  l_qtype_in      VARCHAR2(12) := 'INBOUND';
  l_qtype_out     VARCHAR2(12) := 'OUTBOUND';
BEGIN
  l_options  := sys.mgw_properties(sys.mgw_property('MQ_SendExit','CDEMQDEVSendExit'));
  l_prop     := sys.mgw_mqseries_properties.construct();
  l_prop.max_connections :=1;
  l_prop.queue_manager          := 'MQ_NYCMGW_A7';
  l_prop.hostname               := 'researchportal.adncorp.com';
  l_prop.port                   := 1724;
  l_prop.channel                :=  'OLSS.HIPAA1.UPDATE';
  l_prop.username               :=  'WEBMQUAT';
  l_prop.password               :=  NULL;
  l_prop.inbound_log_queue      :=  funGetQueueName(l_qtype_in);
  l_prop.outbound_log_queue     :=  funGetQueueName(l_qtype_out);
  DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(
                                        LINKNAME => 'MQS_CDEMQUAT',
                                        PROPERTIES => l_prop,
                                        OPTIONS => l_options
                                        );
  DBMS_MGWADM.REGISTER_FOREIGN_QUEUE(NAME => 'OLSS.HIPAA1.UPDATE.REQUESTQ2',
                       LINKNAME => 'MQS_IPC_LINK1',
                       PROVDIER_QUEUE =>  'OLSS.HIPAA1.UPDATE.REQUESTQ2',
                       OPTIONS => MGW_PROPERTIES(MGW_PROPERTY('MQ_openOptions', '1724'),
                       COMMENT => 'Websphere MQ Series Test on  OLSS.HIPAA1.UPDATE.REQUESTQ2'
                       );
EXCEPTION WHEN OTHERS THEN
  DBMS_OUTPUT.put_line(SQLERRM);
END;
```

Using AQ with MQ



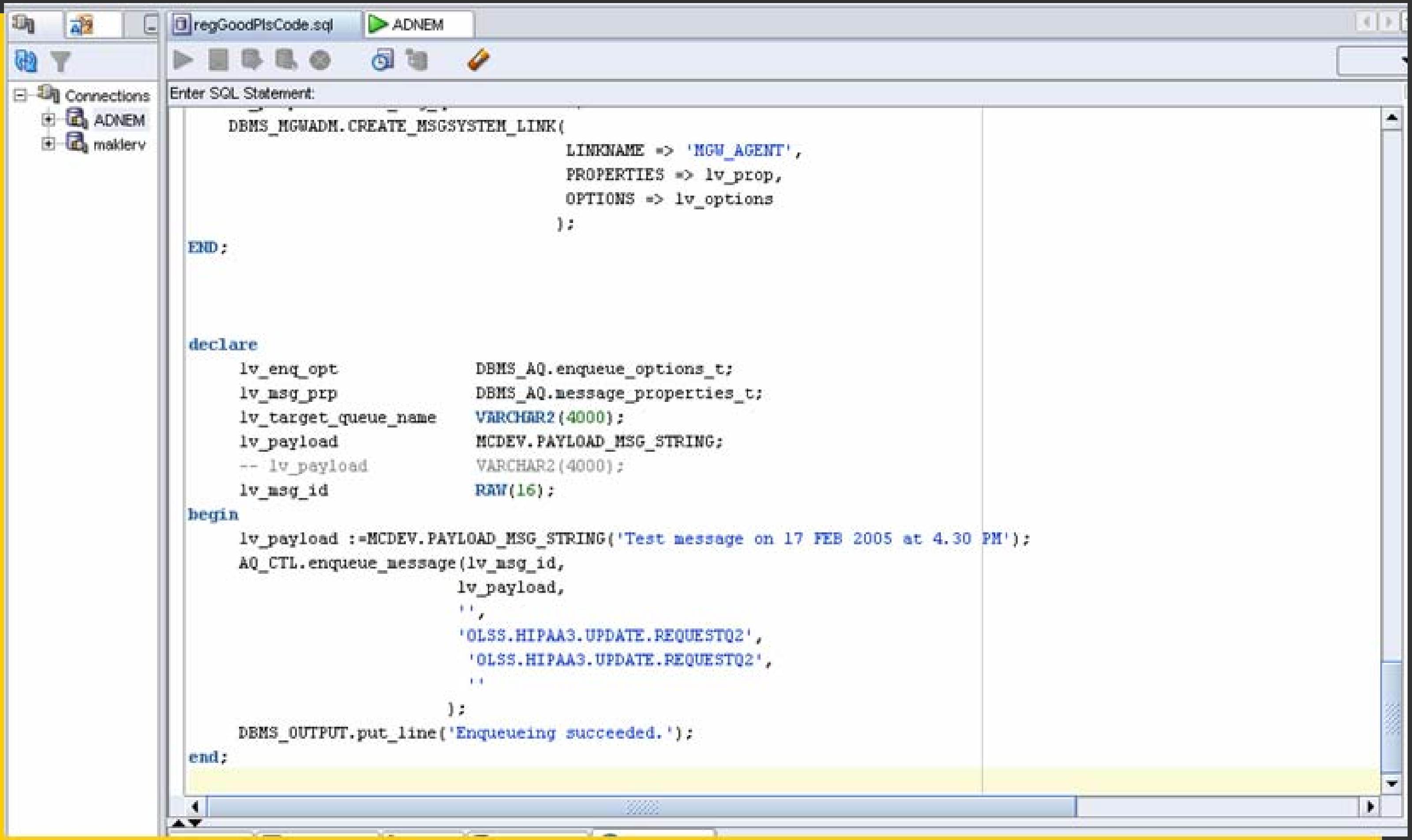
```
DECLARE
   lv_options        sys.mgw_properties;
   lv_prop           sys.mgw_mqseries_properties;
BEGIN
   lv_options  := sys.mgw_properties(sys.mgw_property('MQ_SendExit','ADNMQDEVSendExit'));
   lv_prop     := sys.mgw_mqseries_properties.construct();
   lv_prop.max_connections :=1;
   lv_prop.queue_manager      := 'MQ_NYCGW_A10';
   lv_prop.hostname        := 'portal.adncorp.com';
   lv_prop.port              := 1414;
   lv_prop.channel           := 'MQIIH.ADN1.UPDATE';
   lv_prop.username        := 'WEBMQADN';
   lv_prop.password        := NULL;
   DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(LINKNAME => 'MQS_IPC_LINK1',
                              PROPERTIES => lv_prop,
                              OPTIONS => lv_options
                              );
   DBMS_MGWADM.REGISTER_FOREIGN_QUEUE(NAME => 'MQIIH_ADN1_UPDATE_REQUESTQ2',
                               LINKNAME => 'MQS_IPC_LINK1',
                               PROVDIER_QUEUE =>  'MQIIH.ADN1.UPDATE.REQUESTQ2',
                               OPTIONS => MGW_PROPERTIES(MGW_PROPERTY('MQ_openOptions', '1414'),
                               COMMENT => 'MQ Series Test on  MQIIH.ADN1.UPDATE.REQUESTQ2'
                               );
EXCEPTION WHEN OTHERS THEN
  DBMS_OUTPUT.put_line(SQLERRM);
END;
```

Using AQ with MQ



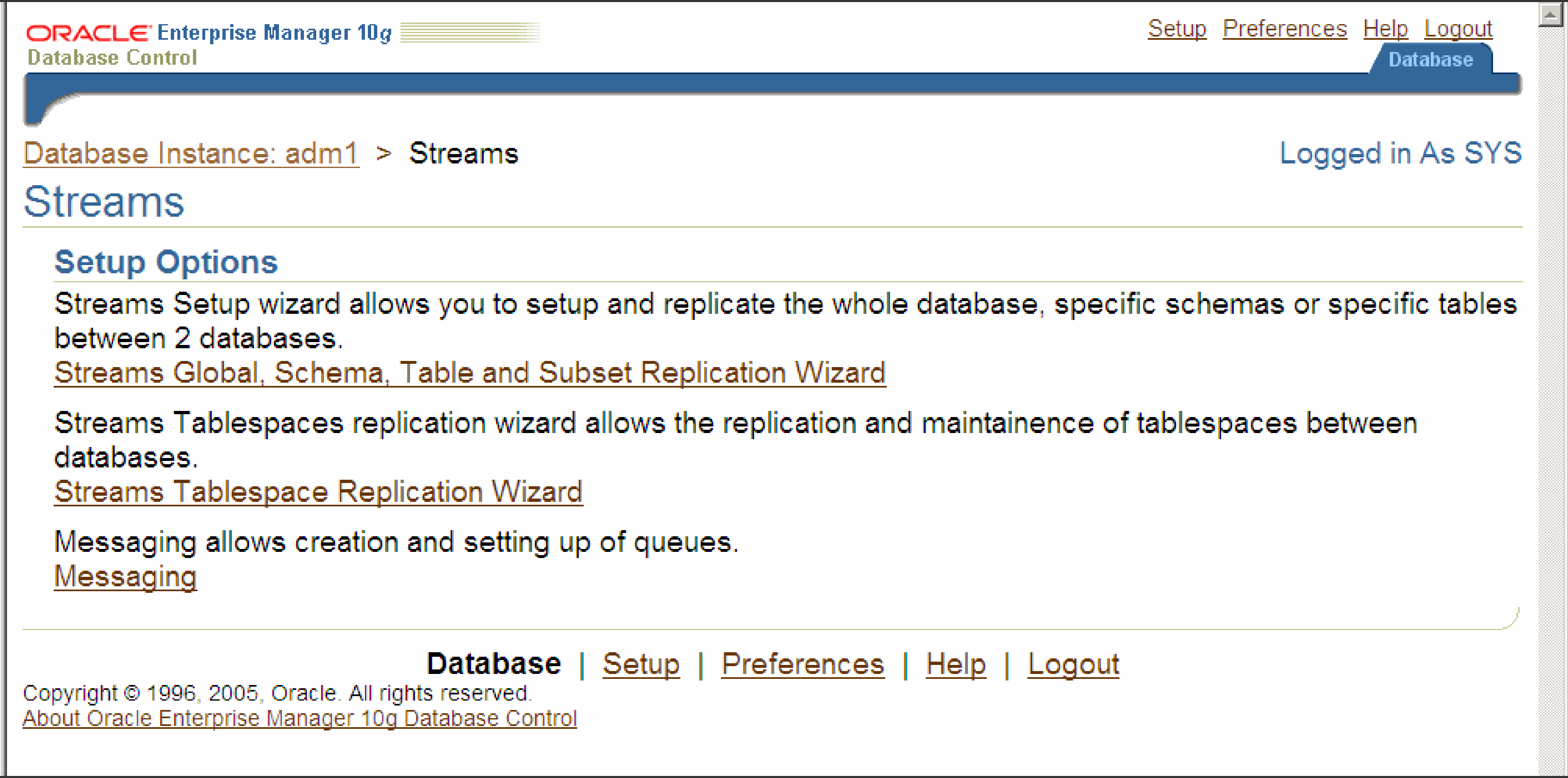
```
DECLARE
    lv_options          sys.mgw_properties;
    lv_prop             sys.mgw_mqseries_properties;
BEGIN
    lv_options  := sys.mgw_properties(sys.mgw_property('MQ_SendExit','CDEMQDEVSendExit'));
    lv_prop     := sys.mgw_mqseries_properties.construct();
    lv_prop.max_connections :=1;
    lv_prop.queue_manager       := 'MQ_ALBWGS_D9';
    lv_prop.hostname          := 'rscdd01.empireblue.com';
    lv_prop.port                := 1414;
    lv_prop.channel             := 'OLSS.HIPAA3.UPDATE';
    lv_prop.username          := 'WEBMQDEV';
    lv_prop.password          := NULL;
--    lv_prop.inbound_log_queue     := 'OLSS.HIPAA3.UPDATE.REQUESTQ2';
--    lv_prop.outbound_log_queue    := 'OLSS.HIPAA3.UPDATE.REQUESTQ2';
    DBMS_MGWADM.REGISTER_FOREIGN_QUEUE(NAME => 'OLSS.OLSSHIP3REQ2',
                                       LINKNAME => 'MGW_AGENT'
                                       );
EXCEPTION WHEN OTHERS THEN
   DBMS_OUTPUT.put_line(SQLERRM);
END;

DECLARE
    lv_options          sys.mgw_properties;
    lv_prop             sys.mgw_mqseries_properties;
BEGIN
    lv_options  := sys.mgw_properties(sys.mgw_property('MQ_SendExit','CDEMQDEVSendExit'));
    lv_prop     := sys.mgw_mqseries_properties.construct();
```

Using AQ with MQ



```
    DBMS_MGWADM.CREATE_MSGSYSTEM_LINK(
                                      LINKNAME => 'MGW_AGENT',
                                      PROPERTIES => lv_prop,
                                      OPTIONS => lv_options
                                     );
END;



declare
     lv_enq_opt              DBMS_AQ.enqueue_options_t;
     lv_msg_prp              DBMS_AQ.message_properties_t;
     lv_target_queue_name    VARCHAR2(4000);
     lv_payload              MCDEV.PAYLOAD_MSG_STRING;
     -- lv_payload           VARCHAR2(4000);
     lv_msg_id               RAW(16);
begin
     lv_payload :=MCDEV.PAYLOAD_MSG_STRING('Test message on 17 FEB 2005 at 4.30 PM');
     AQ_CTL.enqueue_message(lv_msg_id,
                            lv_payload,
                            '',
                            'OLSS.HIPAA3.UPDATE.REQUESTQ2',
                             'OLSS.HIPAA3.UPDATE.REQUESTQ2',
                             ''
                           );
     DBMS_OUTPUT.put_line('Enqueueing succeeded.');
end;
```

Using Database and Grid Control



Database Instance: adm1 > Streams

Logged in As SYS

Streams

Setup Options

Streams Setup wizard allows you to setup and replicate the whole database, specific schemas or specific tables between 2 databases.
Streams Global, Schema, Table and Subset Replication Wizard

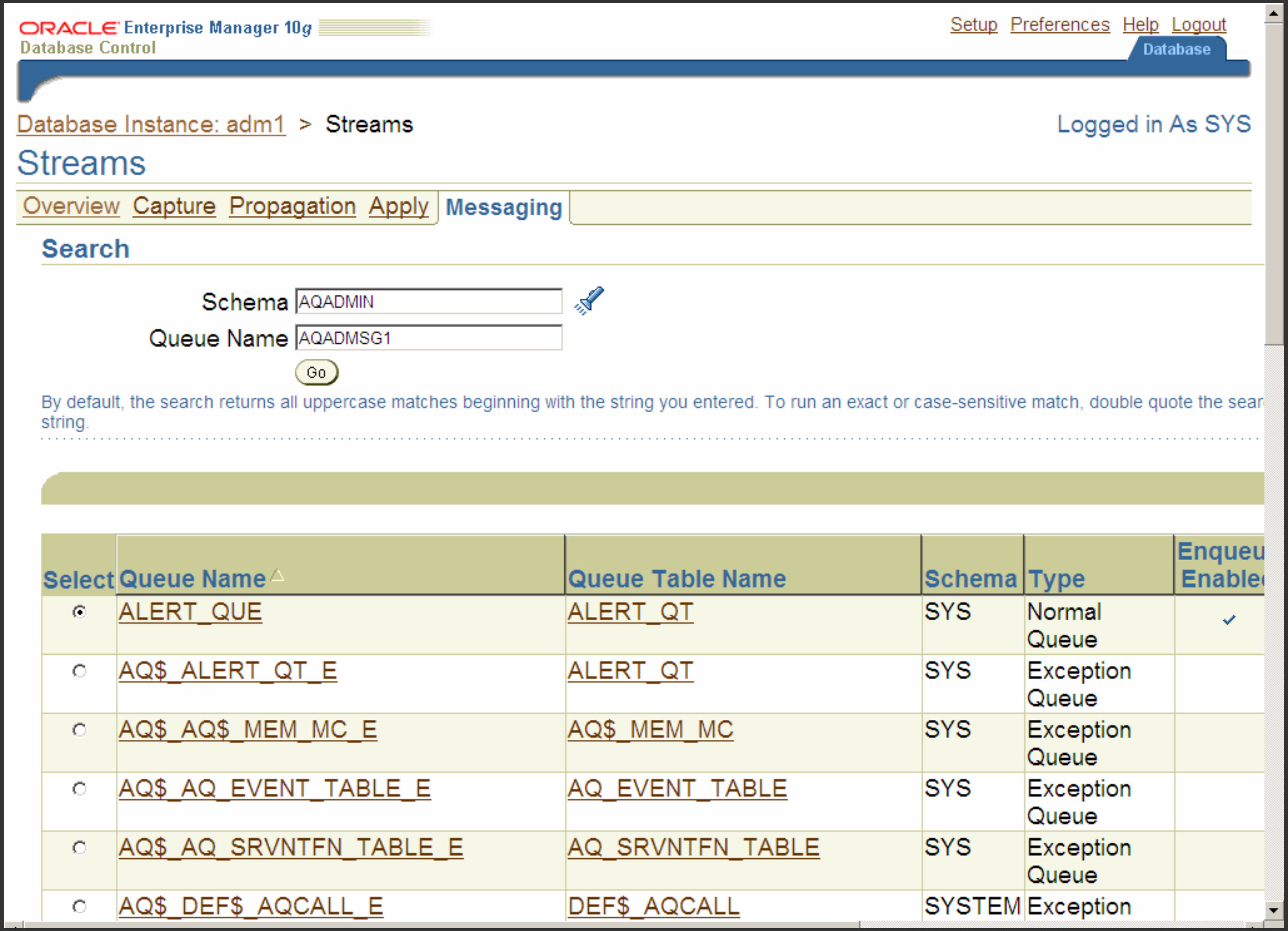
Streams Tablespaces replication wizard allows the replication and maintainence of tablespaces between databases.
Streams Tablespace Replication Wizard

Messaging allows creation and setting up of queues.
Messaging

Database | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Database Control

Using Database and Grid Control



ORACLE Enterprise Manager 10*g*
Database Control

Setup Preferences Help Logout

Database

Database Instance: adm1 > Streams

Logged in As SYS

Streams

Overview Capture Propagation Apply **Messaging**

Search

Schema AQADMIN

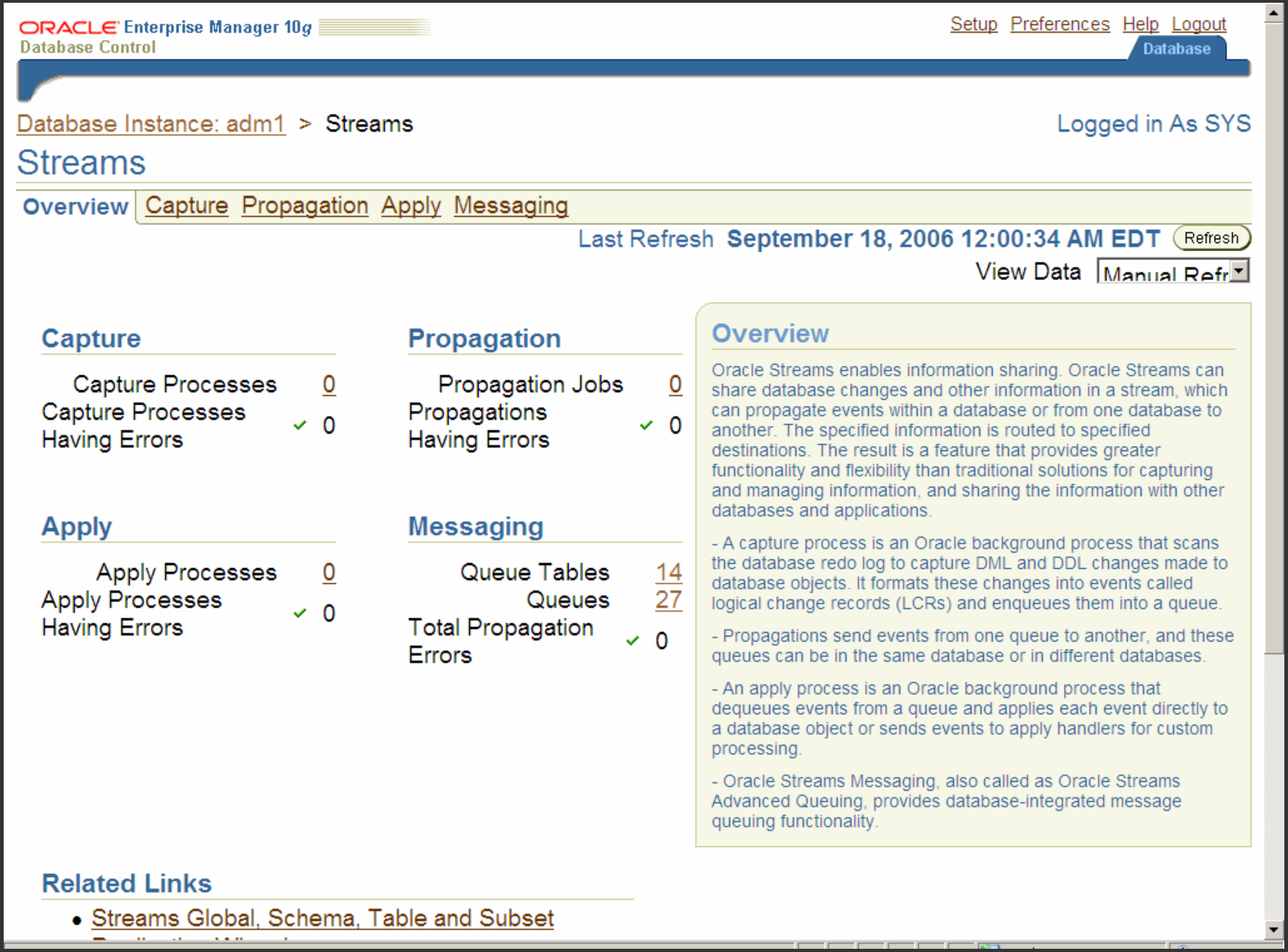
Queue Name AQADMSG1

Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the sear
string.

Select	Queue Name	Queue Table Name	Schema	Type	Enqueu Enable
●	ALERT_QUE	ALERT_QT	SYS	Normal Queue	✓
○	AQ$_ALERT_QT_E	ALERT_QT	SYS	Exception Queue	
○	AQ$_AQ$_MEM_MC_E	AQ$_MEM_MC	SYS	Exception Queue	
○	AQ$_AQ_EVENT_TABLE_E	AQ_EVENT_TABLE	SYS	Exception Queue	
○	AQ$_AQ_SRVNTFN_TABLE_E	AQ_SRVNTFN_TABLE	SYS	Exception Queue	
○	AQ$_DEF$_AQCALL_E	DEF$_AQCALL	SYSTEM	Exception	

Using Database and Grid Control



Database Instance: adm1 > Streams

Logged in As SYS

Streams

Overview Capture Propagation Apply Messaging

Last Refresh **September 18, 2006 12:00:34 AM EDT** Refresh

View Data Manual Refr

Capture

Capture Processes	0
Capture Processes Having Errors	✓ 0

Propagation

Propagation Jobs	0
Propagations Having Errors	✓ 0

Apply

Apply Processes	0
Apply Processes Having Errors	✓ 0

Messaging

Queue Tables	14
Queues	27
Total Propagation Errors	✓ 0

Overview

Oracle Streams enables information sharing. Oracle Streams can share database changes and other information in a stream, which can propagate events within a database or from one database to another. The specified information is routed to specified destinations. The result is a feature that provides greater functionality and flexibility than traditional solutions for capturing and managing information, and sharing the information with other databases and applications.

- A capture process is an Oracle background process that scans the database redo log to capture DML and DDL changes made to database objects. It formats these changes into events called logical change records (LCRs) and enqueues them into a queue.

- Propagations send events from one queue to another, and these queues can be in the same database or in different databases.

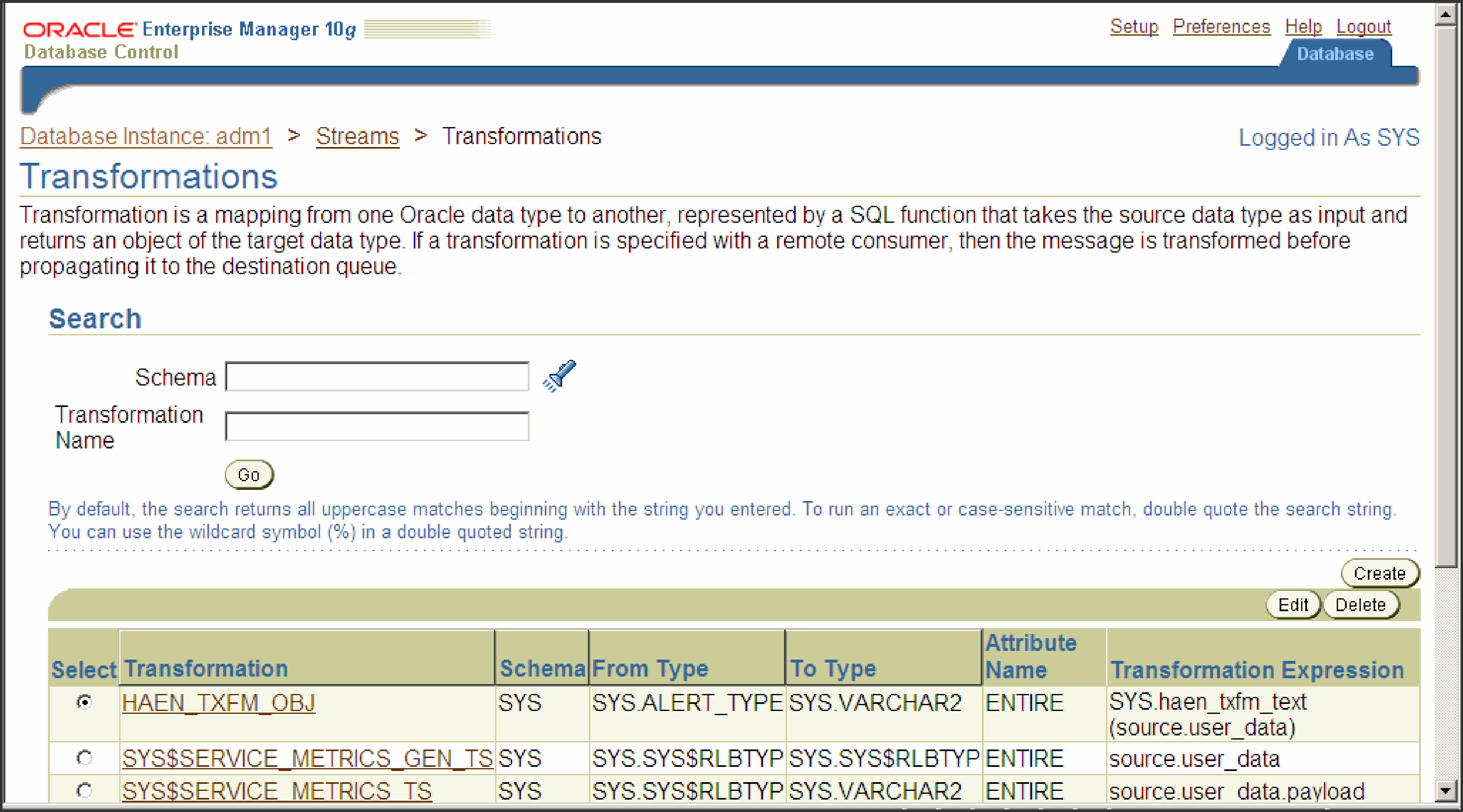
- An apply process is an Oracle background process that dequeues events from a queue and applies each event directly to a database object or sends events to apply handlers for custom processing.

- Oracle Streams Messaging, also called as Oracle Streams Advanced Queuing, provides database-integrated message queuing functionality.

Related Links

- Streams Global, Schema, Table and Subset

Using Database and Grid Control



Database Instance: adm1 > Streams > Transformations

Logged in As SYS

Transformations

Transformation is a mapping from one Oracle data type to another, represented by a SQL function that takes the source data type as input and returns an object of the target data type. If a transformation is specified with a remote consumer, then the message is transformed before propagating it to the destination queue.

Search

Schema:

Transformation Name:

Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Create

Edit Delete

Select	Transformation	Schema	From Type	To Type	Attribute Name	Transformation Expression
●	HAEN_TXFM_OBJ	SYS	SYS.ALERT_TYPE	SYS.VARCHAR2	ENTIRE	SYS.haen_txfm_text (source.user_data)
○	SYS$SERVICE_METRICS_GEN_TS	SYS	SYS.SYS$RLBTYP	SYS.SYS$RLBTYP	ENTIRE	source.user_data
○	SYS$SERVICE_METRICS_TS	SYS	SYS.SYS$RLBTYP	SYS.VARCHAR2	ENTIRE	source.user_data.payload

Advanced Strategies

- OCI- and Precompiler-based Custom Implementation
- RAC-Support (best strategy for the large enterprise)
- Message Priority

Advanced Strategies

- Message-Driven Beans and XA Support
  - Synchronous distributed transactions
  - Two-phase commit (2PC) implementation
- JMS-based custom implementation with Connection and Context Factory
- Protocol support (LDAP, and SMTP, SNMP APIs)

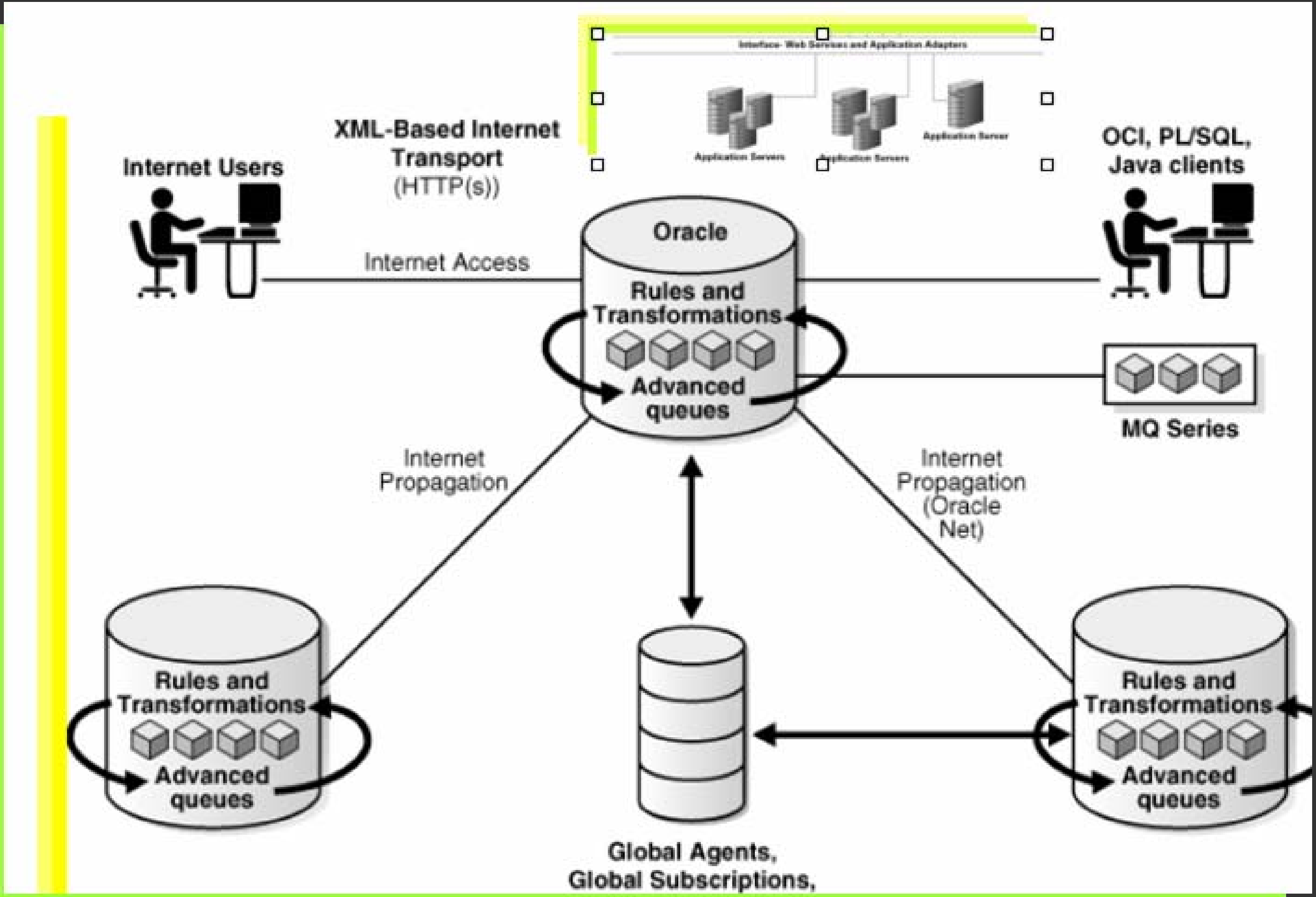
Advanced Strategies

- Buffered messaging, a new feature in Oracle Streams AQ 10g Release 2 (10.2), combines the rich functionality that this product has always offered with a much faster queuing implementation. Buffered messaging is ideal for applications that do not require the reliability and transaction support of Oracle Streams AQ persistent messaging.

Advanced Strategies

- Buffered messaging is faster than persistent messaging, because its messages reside in shared memory. They are usually written to disk only when the total memory consumption of buffered messages approaches the available shared memory limit.

Advanced Strategies



Interface- Web Services and Application Adapters
Application Servers
Application Servers
Application Server
XML-Based Internet Transport (HTTP(s))
Internet Users
OCI, PL/SQL, Java clients
Internet Access
Oracle
Rules and Transformations
Advanced queues
MQ Series
Internet Propagation
Internet Propagation (Oracle Net)
Rules and Transformations
Advanced queues
Rules and Transformations
Advanced queues
Global Agents, Global Subscriptions,

Managing Encryption

- Asymmetric authentication via PKI
  - The producer application encrypts the message payload prior to enqueuing.
  - The consumer application knows the key and decrypts the message.

Managing Encryption

- The approach is also valid for intermediate or repeating queues under the SOA infrastructure, in conjunction with service requestor and receiver, accordingly.
- Encryption can be congruent with payload transformation.

Industries of Application

- Financial Sector
  - Banking
  - Trading
- E-Business (SCM, e.g., B2B transactions)
- E-Business (CRM, e.g., Order Entry)
- Direct Marketing

Related Technologies

- SOA
- Web Services Security and Transaction
- Oracle Streams
- Oracle Advanced Replication
- RPC

Related Technologies

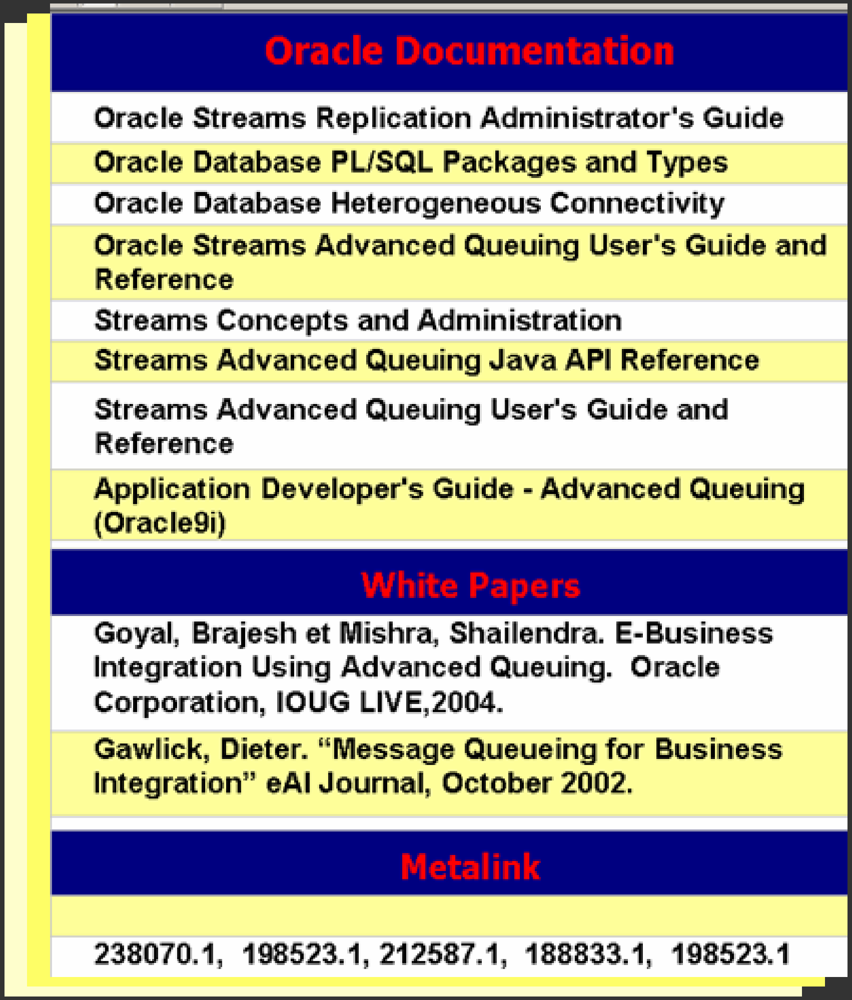
- **AMQP (competitor).**

  In the news: LONDON - June 20, 2006 - JPMorgan Chase & Co., Cisco Systems, Envoy Technologies, Inc., iMatix Corporation, IONA® Technologies, Red Hat, Inc., TWIST Process Innovations, and 29West, today announced the formation of the AMQP (Advanced Message Queuing Protocol) Working Group and an effort by its members to create a new specification for defining and developing messaging infrastructure that is technology agnostic, standards-based, open and interoperable. The AMQP is a binary level protocol that is divided into two layers and designed with a flexible, plug-in architecture. Both the functional layer and the transport layer can be easily evolved to enable AMQP to respond to changing technology requirements.

Strategic Group Partners

- Products working with Oracle Procedural Gateways
  - IBM Websphere MQ (formerly MQ Series)
  - Microsoft MSMQ
  - Tibco Rendez-Vous

Available Literature



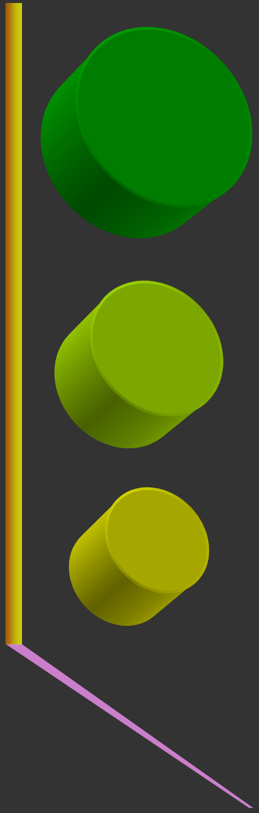
Oracle Documentation
Oracle Streams Replication Administrator's Guide
Oracle Database PL/SQL Packages and Types
Oracle Database Heterogeneous Connectivity
Oracle Streams Advanced Queuing User's Guide and Reference
Streams Concepts and Administration
Streams Advanced Queuing Java API Reference
Streams Advanced Queuing User's Guide and Reference
Application Developer's Guide - Advanced Queuing (Oracle9i)
White Papers
Goyal, Brajesh et Mishra, Shailendra. E-Business Integration Using Advanced Queuing. Oracle Corporation, IOUG LIVE,2004.
Gawlick, Dieter. “Message Queueing for Business Integration” eAI Journal, October 2002.
Metalink
238070.1, 198523.1, 212587.1, 188833.1, 198523.1

Envisioning AQ Future

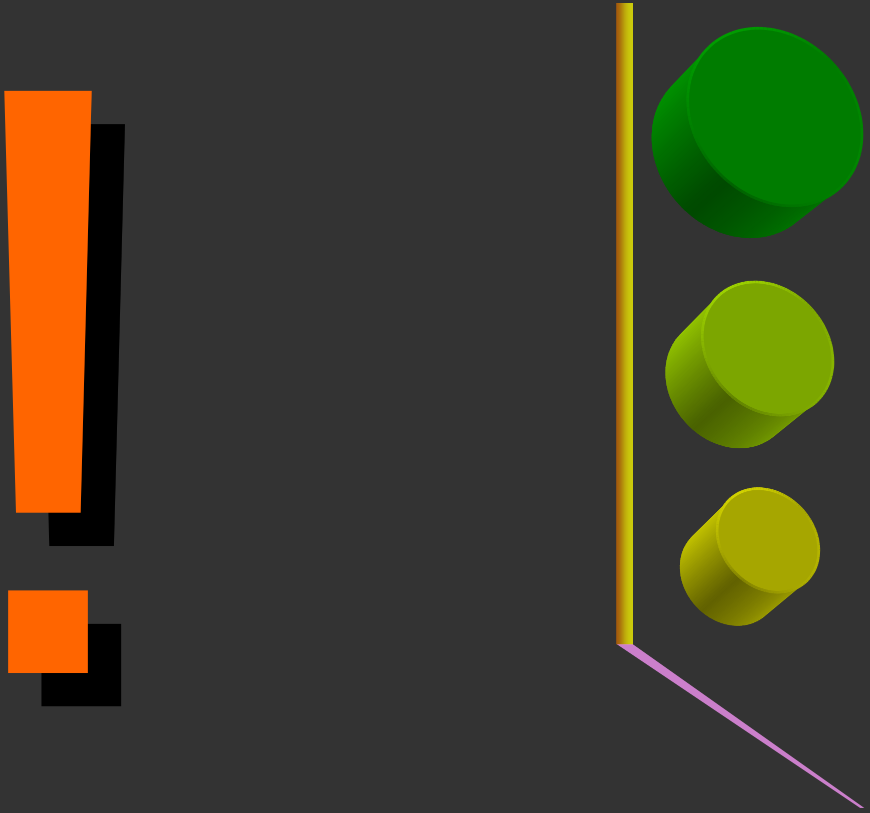
- Who
- Where
- When
- Extent
- How
- How much

Concluding Remarks

- Future of information integration
- Importance of protocol standards
- Information Privacy
- Vendor interoperability
- Business Operational effectivenes.



Demonstration



Discussion

AQ

Q/A

?