

Banner Technical Introduction to Banner Administration Training Workbook

Release 8.0 - April 2008

Updated 4/29/2008



SUNGARD HIGHER EDUCATION

What can we help you achieve?

SunGard Higher Education
4 Country View Road
Malvern, Pennsylvania 19355
United States of America
(800) 522 - 4827

Customer Support Center website
<http://connect.sungardhe.com>

Distribution Services e-mail address
distserv@sungardhe.com

Other services

In preparing and providing this publication, SunGard Higher Education is not rendering legal, accounting, or other similar professional services. SunGard Higher Education makes no claims that an institution's use of this publication or the software for which it is provided will insure compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization's own choosing.

Trademark

Without limitation, SunGard, the SunGard logo, Banner, Campus Pipeline, Luminis, PowerCAMPUS, Matrix, and Plus are trademarks or registered trademarks of SunGard Data Systems Inc. or its subsidiaries in the U.S. and other countries. Third-party names and marks referenced herein are trademarks or registered trademarks of their respective owners.

Revision History Log

Publication Date	Summary
4/30/2008	New version that supports Banner General 8.0 software.

Notice of rights

Copyright © SunGard Higher Education 2005-8. This document is proprietary and confidential information of SunGard Higher Education Inc. and is not to be copied, reproduced, lent, displayed or distributed, nor used for any purpose other than that for which it is specifically provided without the express written permission of SunGard Higher Education Inc.

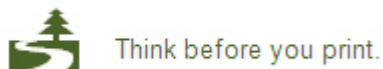


Table of Contents

Introduction	6
Oracle Components	8
Oracle Overview	9
Internet Native Banner – INB	10
Self Service Banner – SSB	12
Web Server Software	14
Self Service Banner - Connection URL	15
DAD (Database Access Descriptor)	17
Oracle Software and Directory Structures	19
The OFA Standard	21
Oracle Installers	25
RDBMS – Tools	26
SQL *Plus.	27
RDBMS - OS Startup/Shutdown	29
Oracle Enterprise Manager	30
Oracle DBA Utilities.....	32
Oracle DBA Utilities.....	33
RDBMS - Network.....	35
The Banner Pieces	37
Banner Overview	38
Banner Components	39
Banner File Locations.....	40
Banner Basics - Software	47
Banner Basics – Product Owners	52
Banner Basics – Example	53
Banner Basics – Triggers.....	54
Banner Basics – Procedures	55
Banner Basics – Referential Integrity	56
Banner Installation	58
Banner Installation – Step 1 (Download the Software)	61
Banner Software Directories.....	63
Banner Software Directories.....	64
Banner General Directories.....	65
Banner New Installation – Step 2 (Environment Creation)	68
Banner New Installation – Step 3 (Database Creation)	69
Banner New Installation – Step 4 (Database Object Creation)	70
Banner New Installation – Steps 5 & 6.....	71
Multiple Banner Environments	72
Multiple Banner Databases.....	73
Banner Default Databases.....	74
Banner Environment	76
Banner Database Management.....	78
Memory Management	79
Storage Management.....	80
Disk Storage Management – Banner	81
Object Management – Banner	82
Banner Utilities.....	83

Oracle Fragmentation	85
Oracle Locally Managed Tablespaces	86
Object Management – Banner	87
Compiling Invalid Objects	89
Banner Database Triggers	90
Additional Banner Software	91
Banner Pro*C	92
Banner Pro*COBOL	94
Banner SQL *Plus	95
Banner SQL *Loader	96
Banner - Access Methods	97
Creating an Banner Database	98
Cloning a Database	99
Banner - Backup Strategies	100
Self Service Banner (SSB) Products	101
Internet Native Banner - Parts	109
Internet Native Banner - Installs	110
Starting INB	115
Upgrades	117
Upgrading Banner	118
Customer Support Center	119
Banner Patches and Upgrades	120
Security	124
Security Overview	125
Oracle Access Rights	126
Oracle Roles	127
Oracle Roles	127
Banner Security Model	128
Banner Security – GSASECR	129
Banner Security – GSAVPDI	130
Banner Role-Level Security	131
Banner - Adding Security to Site Processes	132
Custom Code	133
SQL – FAQ#1979	134
SQL – Actual file	136
C programs – FAQ#2587	137
COBOL Programs	138
Forms	139
Reports – FAQ #1582	140
Self Service Banner	141
Adding Processes to Banner Security	142
BANSECR - Tables and Objects	143
Encryption	146
Banner 8.0 Security Enhancements	148

Job Submission.....	150
Job Submission Overview.....	151
Process, Parameter, Printer Definition.....	154
Job Submission - The Process	155
The <i>jobsub</i> User	159
GURJOBS Startup and Shutdown.....	160
Job Submission - Startup and Shutdown	161
Job Submission – Cleanup.....	162
Summary/Review Objectives	163

Introduction



Course goal

The function of this training program is to provide introductory training to designated personnel in the administration of the Banner software environments. At present, the focus of the course is on the PowerPoint presentation, for which this workbook is a supplement.

Banner administration requires core competencies of a very technical nature. This class deals with configuring the hardware and software infrastructure necessary for the Banner application software to run.

Class participants should leave with the ability to prepare for future maintenance tasks of the Banner databases.

Course objectives

At the end of this course, participants will be able to

- understand the basis of Banner
- understand the Self Service Banner (SSB) and Internet Native Banner (INB) hardware and software architectures
- locate Banner source code
- manage the Banner system
- integrate Banner and other products
- apply a Banner upgrade
- apply Banner security to users and site-created source code
- have working knowledge of Banner Job Submission.

Intended audience

The target audience for this course includes:

- Oracle DBAs
- System Administrators
- Programmer/Analysts
- Network Administrators
- Desktop Support Personnel

Prerequisites

To complete this course, you should have

- Oracle RDBMS installed
- Oracle Application Server installed
- Oracle Forms and Reports installed
- Banner installed
- Oracle SQL and PL/SQL training.

Oracle Components



Introduction

This section discusses the basic components of Oracle products and how they relate to Banner.

Objectives

At the conclusion of this section, participants will be able to:

- demonstrate familiarity with Oracle products, such as
 - RDBMS Server
 - Forms and Reports
 - The Application Server
 - Initiator or Sun JVM.

Oracle Overview

Oracle products

Oracle has many software products for information services customers. With Banner, we are concerned with the following four Oracle products:

- RDBMS Server
- Forms and Reports
- The Application Server
- Initiator or Sun JVM

Software

The RDBMS software manages the databases.

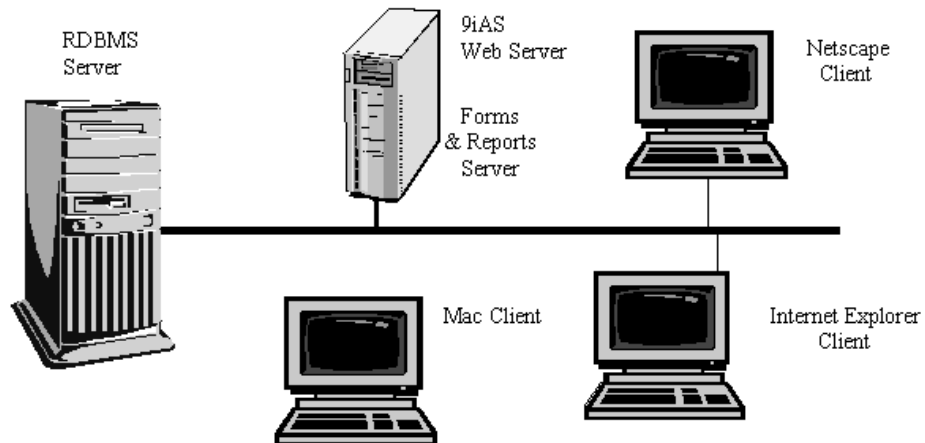
The Oracle Forms and Reports products are the business intelligence/presentation layer for the Banner application.

The Oracle 10g Application Server software provides an internet-based solution that portals Self Service Banner and 'web enables' the Forms and Reports product to present the Banner application over the web.

Initiator is the Java applet that runs inside the client's web browser.

Internet Native Banner – INB

Diagram



Oracle components

- Banner 6.x
 - RDBMS – Oracle 9i Enterprise Edition
 - INB Server – Oracle9iAS v.1.0.2.2.2a Enterprise Edition
 - Forms 6i
 - Reports 6i
- Banner 7.0
 - RDBMS – Oracle 9i Enterprise Edition
 - INB Server – Oracle AS10g Release 1 (9.0.4.x) Forms and Reports CD
 - Forms 9i
 - Reports 9i
- Banner 7.2+
 - RDBMS – Oracle 10g Enterprise Edition
 - INB Server – Oracle AS10g Release 1 (9.0.4.x) or Release 2 (10.1.2.x) Forms and Reports CD
 - Forms 9i or 10g
 - Reports 9i or 10g

Security

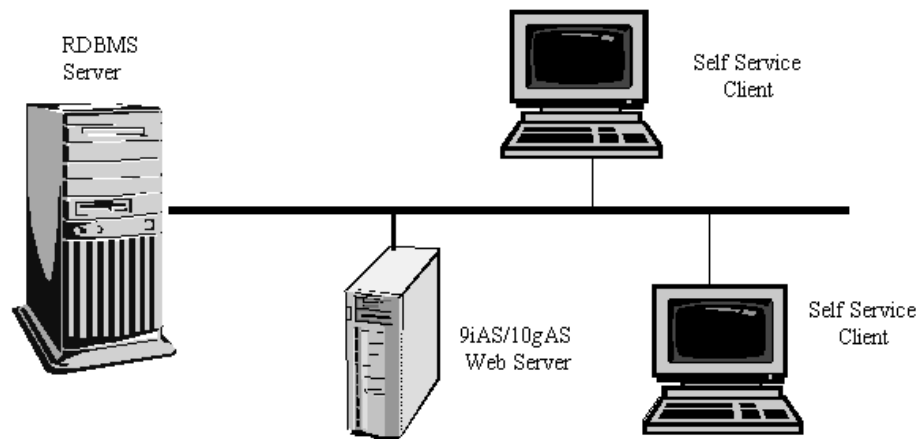
- Oracle user ID/password

Network Transport

- Application Server <-> RDBMS SQL*Net over TCP/IP
- Application Server <-> Client Browser Java over HTTP(S)

Self Service Banner – SSB

Diagram



Oracle Components

- Banner 6.x
 - RDBMS – Oracle 9i Enterprise Edition
 - SSB Server – Oracle9iAS R1/R2 Core Edition or AS10g (9.0.4) WebCache & J2EE CD
 - HTTP Server
 - Database Access Descriptor (DAD)
 - Modplsql cartridge
- Banner 7.x
 - RDBMS – Oracle 9i/10g Enterprise Edition (9i up to Banner 7, 10g from Banner 7.2+)
 - SSB Server – AS10g Release 1 (9.0.4.x) or Release 2 (10.1.2.x) J2EE & WebCache CD
 - HTTP Server
 - PL/SQL Database Access Descriptor (DAD)
 - J2EE and Web Cache only needed for install

Security

- SPRIDEN_ID/PIN
- GOBTAC_EXTERNAL_USER/PIN
- WWW2_USER/password is the single connection to the RDBMS itself.

Network Transport

- Application Server <-> RDBMS SQL*Net over TCP/IP
- Application Server <-> Client Browser HTTP(S) over TCP/IP

Web Server Software

Oracle Internet Application Server

Oracle has changed the name of its web server software product a number of times.

With Oracle8, Oracle Web Application or OWA was born. This name is still reflected within the RDBMS software. OWA utilized the Spyglass listener.

With Oracle9 came Oracle9iAS or internet Application Server software. Oracle9iAS uses Apache for its web listener and required manual editing of its configuration files.

With Oracle 10g, all versions of Oracle AS10g are fully browser-compliant for its configuration through the Enterprise Manager Application Control interface.

Oracle Internet services use Apache. Banner utilizes this in two ways:

- Self Service Banner – uses HTTP core edition
- Internet Native Banner – uses Forms and Reports edition

Self Service Banner

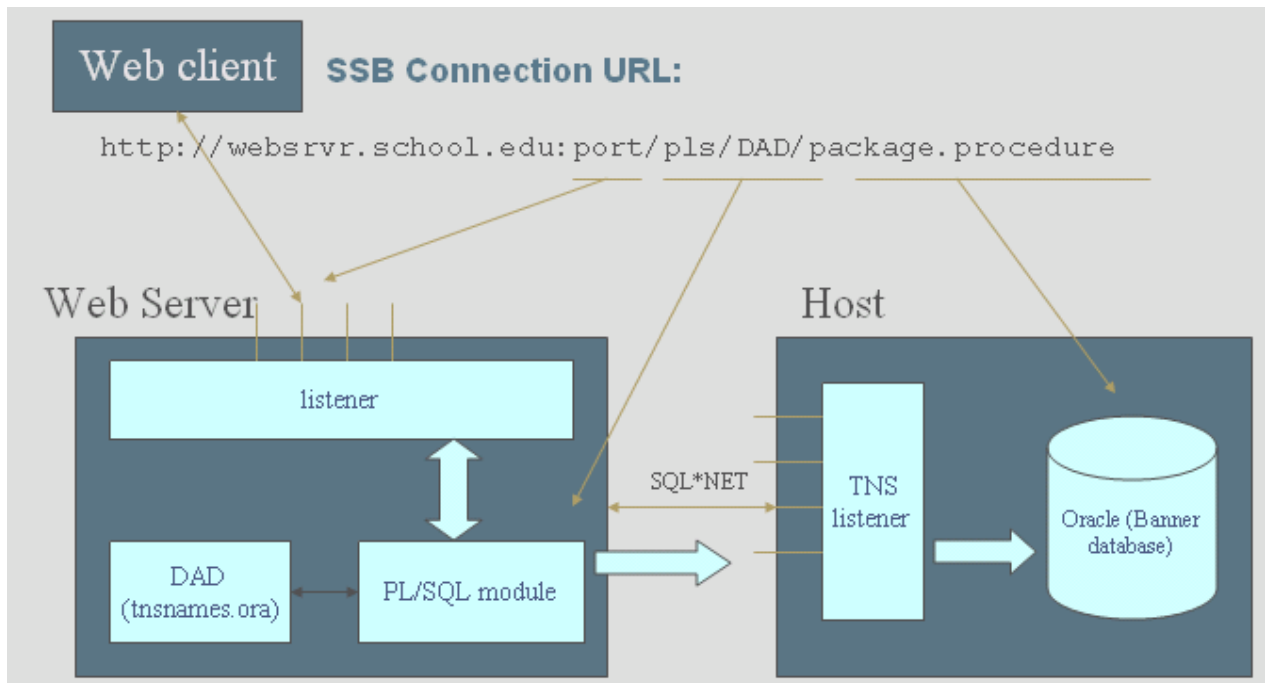
- Uses PL/SQL with a supplied HTTP/HTF toolkit
- Generates Web pages dynamically in the database and serves a static
- Connects to the database with a Database Access Descriptor (DAD)

Internet Native Banner

- Uses servlet and jserv technology
- Requires Oracle Forms and Reports software
- Allows web browser interface with Oracle Forms and Reports by presenting them via Java

Self Service Banner - Connection URL

Diagram



Definitions

The URL is shown on a Self Service Banner page as a menu item, and is implemented as a link behind the menu item.

- The "http://websrvr.school.edu" part is the normal HTML URL portion for the machine.domain name.
- The ":port" part is the designation given in the httpd.conf file for the port to which this VirtualHost is listening.
- The "/pls/SID" is the Database Access Descriptor (DAD), the name that the PL/SQL cartridge or module uses.
- The "/package.procedure" is the name of the procedure that RDBMS calls.

/pls

The "/pls" portion of the DAD was necessary in Oracle9iAS. It is not necessary in Oracle AS10g, but SunGard Higher Education continues to use this as its standard. This makes it easier for DBAs as they upgrade from one version to another.

/package.procedure

The "/package.procedure" when executed as a PL/SQL command first gathers the information from the appropriate Banner tables. The information is then encapsulated within HTML tags using the Oracle OWA packages. The result is passed to the web server as a static HTML page.

DAD (Database Access Descriptor)

Database Access Descriptor

Specifies:

- Uses the TNSNAMES.ORA file alias containing the connect string
 - Host machine name and domain or IP address
 - Port number the Oracle Database listener uses
- Oracle user to connect to database must be defined
 - This is not the end user (SPRIDEN_ID)
 - Usually this is `www2_user`, but may be `www_user` depending on the version of SSB you first installed
 - Changes to the password in RDBMS must also be changed in the DAD and vice versa. There is no automatic mechanism for this like in the older OWA software
- The DAD configuration is contained in the `dads.conf` file

Exercise 1

Where is your web server's documentation?

How do you access it?

What other documentation do you have for Web Applications?

What other resources are there?

Oracle Software and Directory Structures

Database server

- RDBMS
- OCI
- Pro*COBOL
- Pro*C
- Oracle Net Services
- Oracle Enterprise Manager

Application server

- SSB
 - AS10g Release 1 (9.0.4.x) or Release 2 (10.1.2.x)
- INB
 - AS10g Release 1 (9.0.4.x) or Release 2 (10.1.2.x) – Banner 7
 - Forms and Reports 9i – Banner 7
 - Forms and Reports 10g – Banner 7.2+

Client

- HTML Compliant Web Browser with JavaScript support
- INB Only - Java Virtual Machine (Sun JVM or Oracle Jinitiator)

Directory structure standard

SunGard Higher Education has tried to implement a standard directory structure across the RDBMS, INB, and SSB servers. This standard tries to follow the OFA standard that Oracle advocates.

Oracle standard

Oracle software:

/u01/app/oracle/product/version	
/u01/app/oracle/product/10.1.0	<i>RDBMS server</i>
/u01/app/oracle/product/9.0.4	<i>SSB Server</i>
/u01/app/oracle/product/9.0.4	<i>INB Server</i>

Banner standard

Banner software:

/u01/app/sghe/product	
/u01/app/sghe/banseed	<i>RDBMS server</i>
/u01/app/sghe/ssb	<i>SSB Server</i>
/u01/app/sghe/inb	<i>INB Server</i>

The OFA Standard

Docs

A directory structure developed by Cary V. Millsap

The Optimal Flexible Architecture

- A careful naming strategy to eliminate data administration problems
- Allows for multiple versions of Oracle by providing multiple ORACLE_HOME directories
- The original OFA recommendations are published in [The OFA Standard: Oracle 8 for Open Systems](#)
- The OFA document for UNIX can be found in Appendix A of the [Administrator's Guide](#)

Issues

The OFA standard came about as a solution model for the following issues:

- Performance bottlenecks caused by poor organization
- Vulnerability to data corruption
- Adequate management and administration of database growth
- Application failures due to segment growth and fragmentation
- Alleviation of switching among multiple Oracle databases

ORACLE_ BASE

ORACLE_BASE is UNIX-based, but the idea is universal.

A mount point *pm*

A standard directory name *h*

The name of the owner of the product *u*

/pm/h/u

Example: /u01/app/oracle is the Oracle software owner home and is referred to as the ORACLE_BASE environment variable (used by Oracle Universal Installer).

ORACLE_ HOME

To fulfill the OFA structure of multiple Oracle homes:

A standard directory name *h*

The version of the software *v*

/pm/h/u/h/v

Example: /u01/app/oracle/product/10.1.0 indicates the start for Oracle 10g, and is referred to as the ORACLE_HOME environment variable.

Multiple homes

ORACLE_BASE serves as a pointer to a root directory for the OUI to start in, while ORACLE_HOME is used to define each co-existing Oracle product version.

The different tiers of Oracle products each have associated ORACLE_HOMEs defined. They have not yet learned to play nicely together on the same machine.

Some versions of the same software cannot be installed in the same OFA tree:

- Older versions of Oracle were not 'multi-homed'
- Forms and Reports 4.0 through 6i

OFA Standard and Banner

The placement of Banner system is structured to follow OFA, except for one difference. The database tablespace files are not in the expected \$ORACLE_BASE/oradata directories. SunGard Higher Education places the oradata directory directly under a mount point, e.g., /u05/oradata.

Exercise 2

What is your ORACLE_BASE?

What is your ORACLE_HOME?

Database server?

INB application server?

SSB application server?

Do you have multiple Oracle versions?

If so, how do you separate the versions?

RDBMS and/or OAS?

Oracle Installers

Introduction

Oracle has its own installer programs.

- Oracle Universal Installer (OUI)

There may be different ones for each tier, driven by the version of the Oracle software.

- For the RDBMS server
- For the Application Server
- For the Client

RDBMS – Tools

Administrative Tools

- SQLPLUS
- Oracle Enterprise Manager (OEM)
- SQL Developer
- TOAD
- Patrol

Administrative Utilities

- SQL*Loader
- Import/Export

Introduction

- SQL*Plus has been the interactive tool for accessing Oracle databases
- SQL*Plus enables you to manipulate SQL commands and PL/SQL blocks, and to perform many additional tasks as well
- Make sure that the environment variables are set to point to the proper instance
- The only tool to create a database

Through SQL*Plus, you can

- enter, edit, store, retrieve, and run SQL commands and PL/SQL blocks
- format, perform calculations on, store, and print query results in the form of reports
- list column definitions for any table
- access and copy data between SQL databases
- send messages to and accept responses from an end user
- perform database administration.

Syntax

```
sqlplus /nolog
```

- Starts the application
- Can be used to log in as a privileged user with a password file

```
sqlplus "/ as sysdba"
```
- Uses a privileged set of commands that allow actions such as shutdown and startup
- Allows the user of the DBA group to bypass the sys password

Startup/Shutdown

Command-line control of starting and stopping the database instance.

Startup:

```
sqlplus /nolog      (sqlplus "/ as sysdba")  
connect sys/syspass as sysdba  
startup
```

Shutdown:

```
sqlplus /nolog      (sqlplus "/ as sysdba")  
connect sys/syspass as sysdba  
shutdown normal/immediate/abort/transactional
```

RDBMS - OS Startup/Shutdown

Management

- Startup and shutdown can be managed from an OS level utility, such as crontab, oradim, OUI and OEM
- OS startup and shutdown is through the /sbin/init.d structure (UNIX), service (NT) or systartup.com (OpenVMS)
- Should be coordinated with the startup and shutdown of web services
- Oracle supplies a dbstart and dbshut utility for Unix

Oracle Enterprise Manager

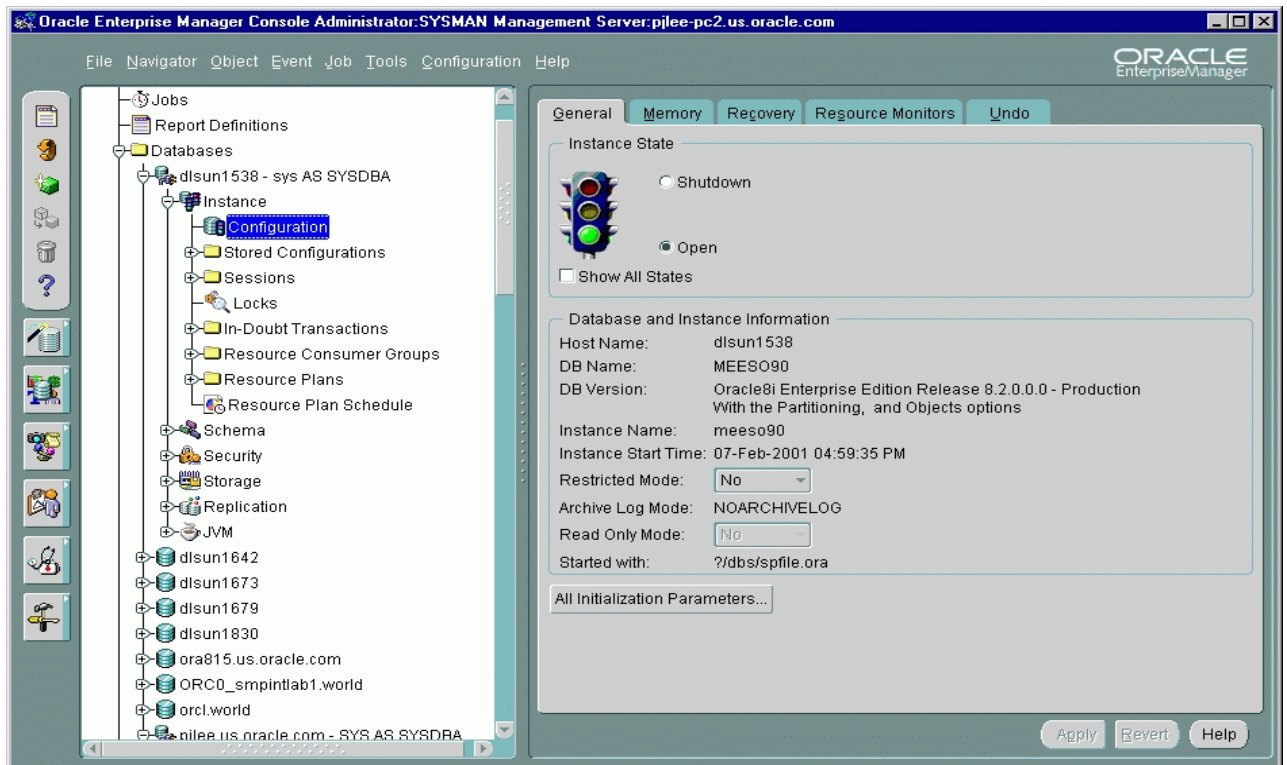
Introduction

- Oracle Enterprise Manager is a product that Oracle has been developing for a long time
- The Enterprise Manager console is a comprehensive tool for database management, configuration and tuning
- This product allows DBAs to manage the Oracle system locally and remotely

Documentation

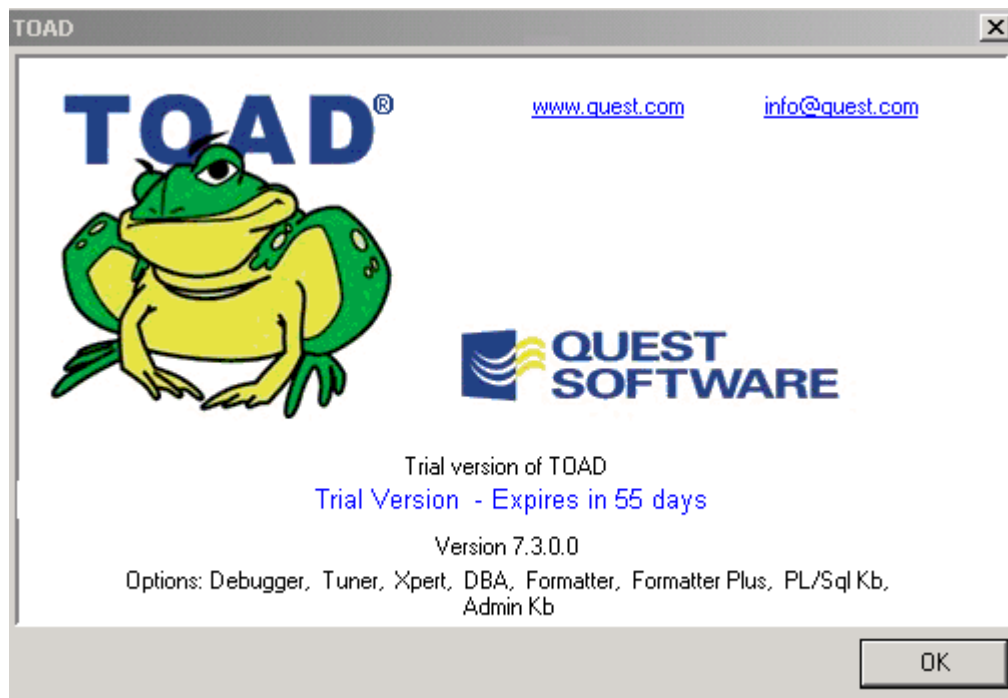
- Oracle Enterprise Manager Administrator's Guide
- Oracle Enterprise Manager Concepts Guide
- Oracle Enterprise Manager Configuration Guide
- Oracle Enterprise Manager Messages Manual
- Oracle Intelligent Agent User's Guide
- Oracle SNMP Support Reference Guide

Instance Manager



TOAD

A product that can provide many of the features of OEM



Oracle DBA Utilities

Utilities

- SQL Loader
Used to import flat files into Oracle objects
- Import
Oracle's binary dump file of one to many objects
- Export
Oracle's binary dump file

SQL*Loader

- SQL*Loader is the Oracle tool that is used to load flat files into Oracle tables
 - Has a powerful data parsing engine which puts little limitation on the format of the data in the data file
 - Can load data from multiple data files during the same load session
 - Can load data into multiple tables during the same load session

Import/Export

Import and Export are a complimentary set of Oracle utilities that are used to import and export data from Oracle databases.

These are conducted while the database is up.

- Import is used to take a previously exported dump file and insert the exported data and/or structure into an Oracle database.
- Exports can be part of your backup strategy.
- A full export can be used for recovery purposes.

Exports can be created with the data from just one table, many tables or the entire database, and may (or not) include indexes and permissions.

Just the database structure can be exported without data.

Caution should be taken with some options such as *rows*, *tables*, *analyze*, *consistent* and *commit*.

See the Oracle Utilities Manual for valuable case studies and available options.

RDBMS - Network

SQL*Net

- SQL*Net was the original proprietary networking protocol supplied with Oracle
- Handled the communication between the RDBMS and Oracle client applications

Oracle Net Services

- The new version of SQL*Net that comes with Oracle 9i and 10G
 - Client to server
 - Server to server
- Routing through a *connection descriptor*
- *Listeners* handle the connection requests
- Configuration can be done through a GUI tool (Configuration Assistant)
- Now supports LDAP
 - Oracle Internet Directory (OID)

Oracle Net Descriptor

- Connection descriptor
- Network route by *protocol address*
- Service name defines the requested service usually a *global database name*
- A *connect string* is built from the above information and passed on the network to the appropriate listener

Descriptor example

Local naming example

```
CONNECT
scott/tiger@(description=(address=(protocol=tcp)
                             (host=myserver)
                             (port=1521))
(connect_data=(service_name=seed.uco.edu)))
```

The Banner Pieces



Introduction

This section discusses the components of Banner in detail.

Objectives

At the conclusion of this section, participants will be able to

- discuss Banner's components
- discuss Banner installation procedures
- manage databases, storage, memory, and objects in Banner
- create and clone Banner databases

Banner Overview

Introduction

Banner is a mature product.

The best you could say about Banner in the early 1990's was that it worked as it said it would. There were enormous gaps in the functional side, though. An example was Credit Evaluation; it was functional at one client site, but could not be made to work for more complex requirements.

In answer to this, SunGard Higher Education built the Curriculum, Advising and Program Planning (CAPP) module in Banner 3.0.

Due to the original version having been built in the late 1980s, the database used was Oracle 5.x. There were a number of features that just did not appear until later versions of Oracle. This caused some items in the database to appear as poor coding practices. An example of this might be the use of tables to hold sequence numbers instead of using Oracle database sequences like the SOBSEQN table.

Integration

One of the design criteria for the original Banner software was to integrate the Student, Finance, HR, Financial Aid, and Development modules. Part of this is accomplished by using one identifier across all modules (PIDM). Another is to clear interfaces between the modules, e.g., the GURFEED table for financial transactions.

Growth

Since the original introduction of Banner, it has gotten a lot bigger. This includes functionality as well as sheer size. Banner is probably one of the most complex Oracle applications in the market place.

The complexity of Banner goes beyond the internal functionality. The additional features provided by third-party software providers have extended the reach of Banner beyond anything originally envisioned.

Banner Components

Documentation set

Banner has several different kinds of documentation. These provide help for different classes of users at different levels. Two examples at different extremes might be the Bookshelf PDF files and the Entity Relationship Diagrams (ERDs).

Banner basics

This is such an integral part of working with Banner that it must be revisited. You must know as much as possible about an object in the Banner software just by its name. Knowing the naming standards gives you a gestalt approach to working with Banner.

This section will discuss naming standards, directory structures, etc.

Software

Once you know how software objects are named you can tell where to find the source code for the object. You should also know which server uses that software.

Database

Objects stored in the database follow naming conventions as well. Knowing these will help you look for them and understand how they work together.

Directory structures

Knowing the placement of the Banner source code will help you in your own future code development. It is relatively easy to manage local source code just by the use consistent directory structures. Version control, of course, is a local matter.

The Banner Installation

Going through this process better informs you of how and why items are placed.

Banner File Locations

Banner files

Location of Banner files in the architecture

- On the RDBMS server

The RDBMS server is the source code repository. All upgrades to the Banner software start on the database server. This should include your source code as well.

- On the Client

Depending on which parts of Banner you use, e.g., spreadsheet based budgeting or a third party reporting tool, there will be no client/server software. What about developers creating Forms and Reports, though? The source code will have to reside on the clients for this kind of development.

- On the Application server (Internet Native Banner)

Copying the forms and reports source code is necessary for generating executables. Who does this? When does this happen?

- On the Web server (Self Service)

Some software must also be moved to the SSB server. Unlike the INB server, though, this is not really source code for compiled objects. These objects are for Self Service help pages.

Help within Banner

- Bookshelf
 - These files contain the basis for Banner training and reference material.
- Embedded help in the database
 - You should have already seen this demonstrated in the Banner General Technical class.
- Online help
- Self Service help
- Table/column comments
- GURPDED

Banner Documentation set

Banner features a documentation set in PDF format.

- Install Guides
- Release Guides
- Upgrade Guides
- Supporting documents
- A set of Technical Reference Manuals

Documentation installation

This documentation set can be downloaded from the SunGard Higher Education Customer Support Center.

Once the zip file is on your machine and unzipped in the correct directory, you need to run the *unlded* utility. The *unlded* utility will uncompress the trz files into the appropriate subdirectories. The bookshelf.pdf file will be at the root of this directory.

Make sure this directory is on the proper server.

Documentation manuals

- Bdoc7x directory
 - Product manual directories
- Finance7x, advanc7x, student7x, etc
 - Index directory
 - _index7x
 - Technical reference directory
- General7x, advanc7x, arsys7x, finaid7x, finance7x, humres7x, student7x
- Bkshlf7x.pdf
 - Startup page for the Banner Bookshelf
- Security
 - General/security700000trm.pdf must be downloaded from the patch site. It does not come with the complete download
 - This manual does not appear in the index, so the casual user will not find it.

User Manuals

- Available online to all Banner users
- Uses clients' Adobe Acrobat Reader
- User Release Guides
 - Each upgrade contains one, containing the changes for the current release
 - Should be stored on/accessed from a shared file server
 - Initially installed on INB server for quick access by all users
 - Do not support all Acrobat functionality
- Set up through an Banner form GUAUPRF
 - Defines location of Banner Bookshelf
 - Globally defined with special user BASELINE

Technical Manuals

- Technical Reference Manuals
 - Available online to all Banner users
 - Contains the database schematics
 - Additional documents can be found in Extended Search on the Customer Support Center - ERDs
- Great source of reference for
 - Security – GSASECR
 - Job Submission
 - Site-specific change management
 - Process flows

Upgrades

- Includes a set of change documents and source files.
 - Upgrade Guide is the step by step guide to upgrading releases
 - User Release guides
 - Change documents – very valuable for deleted and/or changed data types, objects, and aliases
 - baninst1_70100to70200.pdf
 - PRODUCT_70100to70200.pdf
 - PRODUCT_objects70100.pdf

Exercise 1

What Banner products did you purchase?

What Banner products are installed?

Who has access to the Banner manuals?

Which manual discusses distributed Banner security?

Banner Basics - Software

Naming Conventions

This information is in the Banner Getting Started Guide.

- Product modules
 - Each product has its own code value, e.g., "S" for Student. Some of these codes are not obvious, e.g., "N" for Position Control.
- Product directories
 - Just as each product has its own code value, each product is installed in its own directory. Remember that while Banner is an integrated Enterprise Resource Planning (ERP) tool, each of the main products can be sold separately.
 - These directories make up the software repository
- Software Objects – These are the objects that need to be thought of as the software repository objects. Each of these object types has its own subdirectory.
 - Pro*C source code
 - Pro*COBOL source code
 - Forms binaries (FMB)
 - Reports binaries (RDF)
 - Database object source code
 - Packages, procedures, functions, triggers
 - Views
 - APIs are packages
 - SQL*Plus code not stored in the database

Products

Product modules

- Admin – Administrative software
- Alumni – Advancement software
- Aluweb – Self Service for Advancement software
- Arsys – Accounts Receivable software
- BANINAS – Banner Institutional Needs Analysis System software
- Common – Common software among products
- Facweb – Self Service for Faculty software
- Finaid – Financial Aid software
- Finance – Finance software
- General – General software
- Genweb – Self Service for General software
- Intcomp – Integration component software
- Infoacc – Kiosk software
- Payroll – Human Resources software
- Posnctl – Position Control software
- Payweb – Self Service for Employees software
- Scomweb – Common Faculty/Student Self Service software
- Student – Student software
- Stuweb – Self Service for Student software
- VRBrite – Voice Response using Brite software
- VREpos – Voice Response using Epos software
- VR-Student – Voice Response for Student software
- Wtailor – Web Tailor software

Product directories

Product directories are named after the product.

Each of these directories has a common set of subdirectories.

- c
- cob
- forms
- dbprocs
- views
- plus
- misc
- install

Compiler conventions

There are different naming conventions for each type of software objects within these directories

- Pro*C
 - .pc - Pro*C source file
 - .h - C header file
 - .c - C source file
- Pro*COBOL
 - pco - Pro*COBOL source file
 - cob - Pro*COBOL copylibs

Forms conventions

There are different naming conventions for each type of software objects within these directories.

- Forms
 - .fmb - Forms binary source file
 - .mmb - Menu binary source file
 - .pll - Procedure Library
 - .rdf - Reports binary source file

SQL conventions

There are different naming conventions for each type of software objects within these directories.

- dbprocs
 - xxKxxxx.sql - package source code
 - xxPxxxx.sql - procedure source code
 - xxFxxxx.sql - function source code
 - xxTxxxx0.sql - trigger source code
- views
 - xxVxxxx0.sql - view creation
 - plus - xxxxxxx.sql

Self Service products

There is a difference in the naming conventions used for the Self Service products.

- Dbprocs
 - TWGKxxxx.sql - Web Tailor Global package source code
 - BWAKxxxx.sql/HWAKxxxx.sql - Advancement Self Service package source code
 - BWCKxxxx.sql – Common Self Service package
 - BWFKxxxx.sql – Finance Self Service package source code
 - BWGKxxxx.sql/HWGKxxxx.sql – General Self Service package source code
 - BWLKxxxx.sql – Faculty Self Service package
 - BWPKxxxx.sql/HWPKxxxx.sql - Employee Self Service package source code
 - BWRKxxxx.sql/HWRKxxxx.sql - Finaid Self Service package source code
 - BWSKxxxx.sql/HWSKxxx.sql - Student Self Service package source code

Note: All products have both BW* and HW*. The HW* files are being phased out. Only use BW* unless otherwise instructed.
- Misc (only UNIX or Windows)
 - Compile scripts for Pro*C, Pro*Cobol, Forms and Reports
 - Shell scripts for executables
 - Perl scripts for NT
- Com (only OpenVMS)
 - Compile scripts for Pro*C, Pro*Cobol, Forms and Reports
 - DCL command procedures for executables

Banner Basics – Product Owners

Product Owners and Codes

General	GENERAL	G
Student	SATURN	S
Finance	FIMSMGR	F
AR	TAISMGR	T
Position Control	POSNCTL	N
Payroll	PAYROLL	P
FinAid	FAISMGR	R
Advancement	ALUMNI	A
Self Service	WTAILORE	TW

Additional products

These are the nine basic products a client can purchase. There are a host of other products that SunGard Higher Education continually adds onto this list, but this is the starting point for those products.

A partial listing of these products would include Web Tailor/wtailor/TW, Integration Component/icmgr/IC, Finance Self Service/bsfmgr/FW. A more complete list can be found in the GTVSYSL table.

Banner Basics – Example

Database objects (Tables and views)

Example:

G	L	R	V	R	B	L
1	2	3	4	5	6	7

- 1 - Primary Module (S- Student, G- General, F- Finance)
- 2 - Sub-module (L- Letter generation, G- General Person)
- 3 - Type of object (R- Report, A- Form, R- Repeating table)
- 4-7 - Meaningful name
- If Pos 2 is a T and Pos 3 is a V, then the object is a validation table

Banner Basics – Triggers

Triggers

aT_abcd_XXXXXXXXXXXXXXXXXXXX (Total no more than 29)

- a = Product identifier, i.e. (S) Student, (P) Payroll
- T = Trigger
- abcd = Table name
- XXXXX = Meaningful trigger name

Banner Basics – Procedures

Database

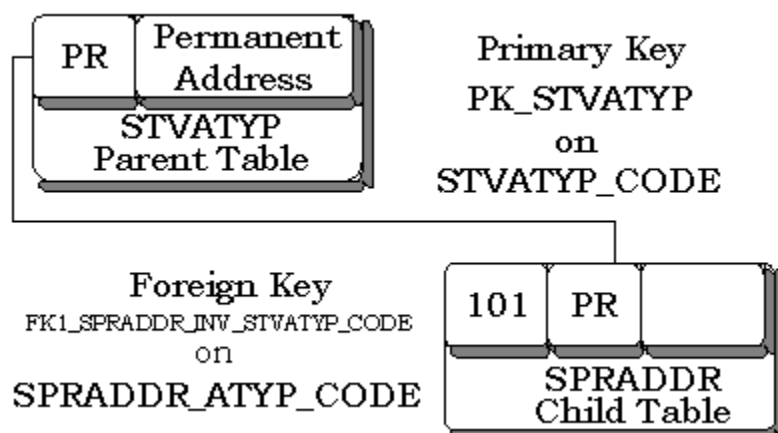
- Procedures and Functions
 - P_XXXXXXXXXXXXXXXXXXXX (Total no more than 29)
 - F_XXXXXXXXXXXXXXXXXXXX (Total no more than 29)
 - xxxxx = Meaningful name (Up to 27 characters)
- Cursors
 - XXXXXXXXXXXXXXXXXXXXXXXXXXXX_C

Banner Basics – Referential Integrity

Referential integrity

- Parent/Child relationships are enforced at the database level
- No child data without a referenced parent key
- No deletion of parent key while child records exist

Example



Details

This example shows the integrity relationship between two tables, STVATYP (the parent table) and SPRADDR (the child table).

STVATYP has the code "PR" in the primary key column, STVATYP_CODE. SPRADDR had the corresponding code of "PR" in the foreign key column, SPRADDR_ATYP_CODE.

Conventions

Note the naming conventions used for the primary key and the foreign key. You should be able to look in the data dictionary to find the necessary columns when either of these constraints is violated.

Exercise 2

What do you know about the column SPRADDR_ATYP_CODE?

The package for Admissions Checklist Processing was accidentally dropped.

How would you restore it?

Which product does BWSKANAM belong to?

Do all of the products contain gifs and icons?

Banner Installation

New Install

When a system is first installed or a new version is released, it is installed from CD-ROM. The CD-ROM contains an Adobe Install guide which specifies:

- System requirements
- Space allocation
- Directory structure
- Environment requirements

C and COBOL must be installed prior to installing along with an empty database. This is usually the client's responsibility to install these compilers. The MicroFocus COBOL compiler requires a key server be running to compile programs. On reboots, this is sometimes overlooked, as you will very rarely compile COBOL programs except for doing Banner upgrades.

New Install - First Database

A database must be created before you can begin.

All tablespaces should be locally managed, autoextend, uniform extent size, and automatic segment space management.

The database must have these tablespaces:

- System (suggested at least 500MB)
- Temp (suggested 200MB)
- UNDO (Default if using DBCA, if not, 200MB)
- Tools (suggested 25 MB)
- Users (suggested 20 MB)
- Development (suggested at least 500MB)
- At least three redo groups

Encryption Key

- CD-ROM of the Banner software contains the database contents and supporting code
 - All software on the CD-ROM is encrypted
 - Encryption Keys are on a sheet of paper shipped with the order.
DO NOT LOSE THIS!
- Software is decrypted and downloaded onto RDBMS server
- This decryption key is not the same as the site specific (explained later in Upgrading Banner)

Environment

- BANNER_ROOT is the variable to list the top level of all Banner software on the server
- BANNER_HOME is a variable used to define the location of an Banner software tree
- Used by Banner software as a base alias to find other Banner software

Installer (baninst)

There is an installer (*baninst*) for the Banner product on the RDBMS machine.

- Implemented differently on each platform
- Modeled off of the old Oracle character mode installer
- Menu interface uses Perl for Windows, shell scripts on UNIX and DCL for OpenVMS

Process

The Banner database software is installed on the RDBMS server in several steps:

- Download Software
- Create Banner Environment
- Create Banner Instance
- Create Banner Database Objects
- Compile Pro*C and Pro*COBOL

Banner Installation – Step 1 (Download the Software)

Components

The first step in the installation process is to download the software to the RDBMS server.

There are four basic components of downloading and preparing to install Banner:

- Identify Products to download
- Define Global Variables
- Download the software
- Build the environment control scripts

Identify products

Identify Products to download (banpswd.shl)

- First, two C programs (crypfile.c and inipassr.c) must be compiled to use in decrypting the Banner products
- Once these are done, the install script will ask for a decrypt password
- This will then build a menu with the appropriate products marked for downloading

Global definitions

Define Global Variables (banprmp.shl)

- BANNER_HOME
- EXE_HOME
- ORACLE_HOME
- ORACLE_SID
- C
- COBOL

Exercise 3

What must be installed before you can start a Banner install?

Do you still have your original install directory?

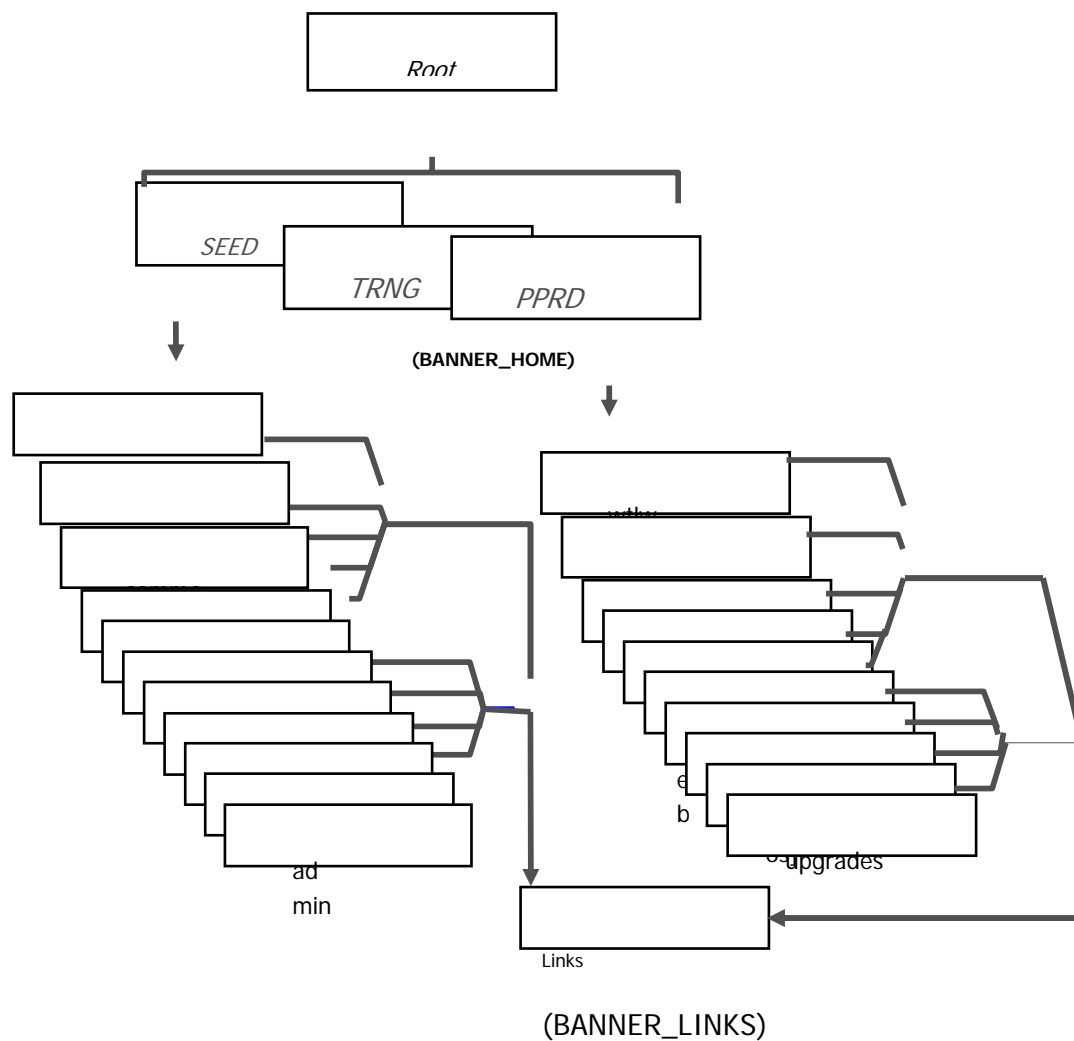
What is a decryption key used for?

What is the name of the Banner installer?

Software download

- Download the software (bansrce.shl)
 - This process will download the software from the CD
 - Decrypt
 - Decompress
 - Populates the following directory structure

Banner Software Directory



The Banner Software Default Directory Structure – example only

Banner Software Directories

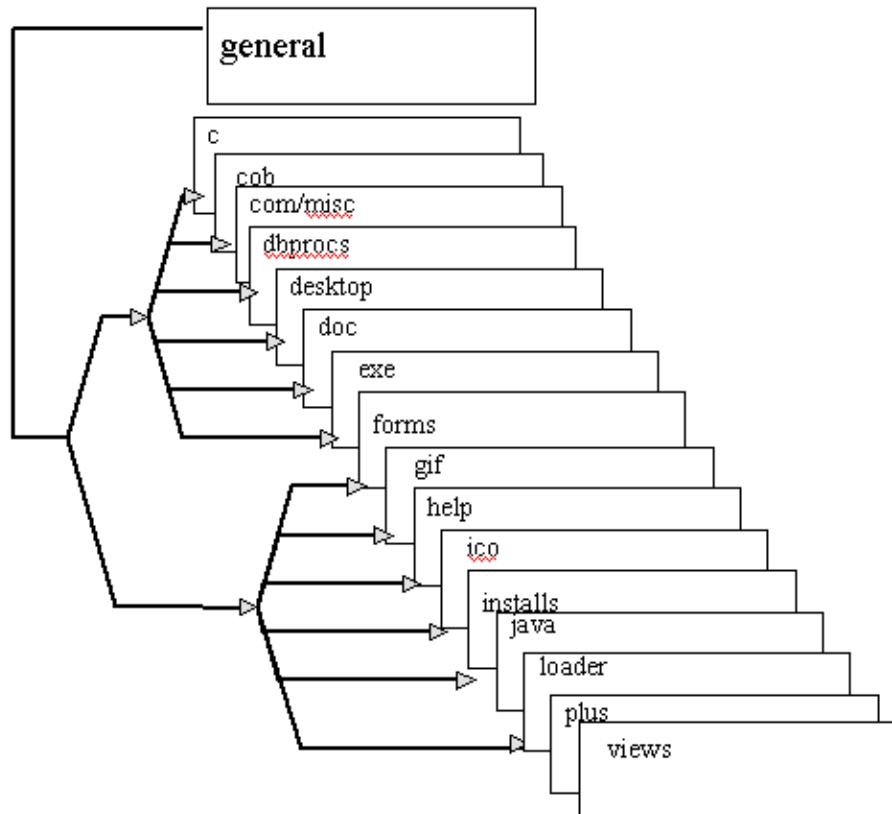
Directories

BANNER_HOME

- Defines the home directory of the Banner products
 - Each product has its own directory
- General is a special product
 - Contains executables and administrative objects
- Common/scomweb is used for cross-product objects
- Upgrades are used to stage Banner upgrades
- Links (UNIX only) directory is used to compensate for older versions of UNIX limit of 255 characters for environment variable
 - It centralizes all of the code in a single code tree version

Banner General Directories

Diagram



The Product Software Default Directory Structure

Banner Product Directories

- C directory contains the Pro*C source
- COB directory contains the Pro*COBOL source and COBOL copybooks
- MISC directory contains the Perl scripts used on the NT platform and Shell scripts for UNIX
 - On VMS, the COM directory contains the DCL scripts
- DBPROCS contains the source for database procedures, functions, triggers, and O:A functions and procedures
- FORMS contains the source for the forms, libraries, menus and reports
- HELP files for the modules forms
- PLUS directory contains SQL*Plus scripts
- LOADER contains any SQL*Loader scripts
- VIEWS contains source for the database views and O:A views
- INSTALL contains the *.sctdmp file of the product

General Directories

General directory contains some special subdirectories:

- EXE contains all the executables for all products
- GIF contains the .gif files used in Banner
- ICO contains the icon files used in Banner
- DESKTOP contains the GODDTOP for spreadsheet budgeting
- EXTRAS contains the scripts for SSO

(may have been replaced by bansso at root level)

Important Users

BANINST1

- Owns all the functions, packages, procedures, and views
- WTAILOR owns the packages for SSB (starting with tw*)

BANSECR

- Owns the Banner security tables and objects
- No other user should have access to BANSECR's objects
- The only exception is for a site-specific BANSECR user

Build environment manipulation scripts

The install process then builds some environment manipulation scripts for managing Banner variables (bansubs.shl):

- banenv - for the Bourne shell
- cbanenv - for the C shell
- *banali - for printer aliasing for the various shells
- Banlogic - for VMS

Registry entries or batch files used for NT

Banner New Installation – Step 2 (Environment Creation)

Environment creation

The next step of the install process will create an Banner environment on the RDBMS server.

- Build the LINKS directory (banlnks.shl –UNIX only)
- Update the environment (bansenv.shl)
- Oraenv is then modified to point to the banenv and cbanenv files (UNIX)
- Oratab is modified with the new database
- Banlogic.com is modified with directory locations (VMS)
- Registry or Batch file banenv.bat (NT)

Banner New Installation – Step 3 (Database Creation)

Database creation

The next set of steps creates the Banner database.

- Uses a pre-created database (bancrdb.shl, bancrtb.shl are provided for reference)
- Creates the Banner accounts (bancrus.shl)

Banner New Installation – Step 4 (Database Object Creation)

Database object creation

The next set of steps in the NEW install will create the database objects.

Note: Review the install document carefully. The DAD user **www2_user** needs to be created before these steps.

- Import Banner tables and sample data (banimp.shl)
- Build Banner procedures and functions (bandbpr.shl)
- Import views and create synonyms (banvwsy.shl)

Banner New Installation – Steps 5 & 6

Compile C & COBOL

- Create make files

In the UNIX environment, SunGard Higher Education has a C program that merges the Oracle sample make file with a template. The template is SunGard Higher Education's best guess at a generic make file. Different versions of UNIX may require slight changes to the make files.

OpenVMS and Windows do not have the variants found in the UNIX world. The compile scripts, therefore, are pre-created.

- Compile General objects

The General product contains the Banner role level security routines. This means that the General product must be compiled first.

- Test the compiles

Run the gjrrpts program to test the C compiles. While this is a fairly simple program, it seems to be a good test model and is available for all products. A more comprehensive test would be to run the posting program in Finance, but that implies that you have purchased Finance.

Testing the COBOL compiles is more complicated. One of the most complex COBOL programs is GLBPARAM.

To test this program, run the GLRVRBL form. Select the Admissions application in this form. Select the first variable in the list (it really doesn't matter which one it is.) Overwrite any character in the where clause with itself, thereby setting the flag that something has changed. Exit the form, and the GLBPARAM program will run in the background.

Multiple Banner Environments

Multiple Banner software environments

- Initial install only builds one code tree
- Need to copy this to another tree to use as an upgrade staging area
- Must manage the access to these different sets of Banner code by using:
 - Shell scripts/registry entries/DCL procedures
 - Banenv/banlogic.com
 - DOS .bat files

Multiple Banner Databases

Multiple Banner instance databases

- Initial install only built one database
- Need to clone this to a training database and a production staging database
- Must manage the access to these different Banner databases by using:
 - Shell files/registry entries/DCL procedures
 - Banenv/banlogic.com
 - DOS .bat files

Banner Default Databases

SEED

- Out-of-the-box Banner
- Used to run new Banner upgrades against first
- Upgrade issues are then SunGard Higher Education issues
- Work out the details before upgrading production database

TRNG

- Used for implementation training
- Out-of-the-box Banner with some additions
 - Training user accounts
 - Training data
- Usually cloned from SEED

PPRD

- Used for implementation and conversion
- Out-of-the-box Banner with some deletions
 - *Most* SEED data is cleaned out of the database
- Used to build the Production database
- Should be backed up as soon as users start populating validation and rule tables

Exercise 4

You've just upgraded Oracle – will you be able to compile Pro*C and Pro*COBOL?

Does the Banner installation itself set up the Job Submission user ID and directory?

What user IDs do your users use when logging into the PPRD database?

What is the first database that is installed?

Banner Environment

Multiple environments and variables

Multiple Banner environments will need both Oracle and Banner variables set.

- .profile or .login calls oraenv (UNIX)
- Login.com calls orauser_SID.com (VMS)

ORAENV sets:	orauser_SID.com sets:
ORAENV_ASK	--
ORACLE_HOME	ORA_DB
ORACLE_SID	ORA_SID
PATH	SQLPATH

VMS

Command procedural calls:

- Banlogic.com
- Orauser.com

NT/Win2000

- Path environment
- Registry

Banner environment variables

Based on the SID, the appropriate environment file is called.

BANENV/Registry sets	BANLOGIC sets
BANNER_HOME	BAN_HOME
BANNER_LINKS	BAN_ROOT
BANNER_ADMIN	<MODULE\$PATH>
EXE_HOME	GEN\$EXE
ORACLE_PATH	ORA_PATH
PATH	SOLPATH

Banner Database Management

Performance

Management of a Banner database involves proactively managing:

- Memory
 - Are your users complaining?
 - Is your system page-faulting?
- Running SQL scripts to maintain the system
- Disk storage
- Fragmentation
- Database objects
- Invalid state

Administration

- Close relationships with the System and Network staff
- Patches and upgrades are maintained
- Backup and recovery in place
- Network access
- It also involves being able to:
 - Create databases
 - Clone databases
 - Upgrade all Oracle software
- "How to have a life and be a DBA also"

Memory Management

Banner

Banner moves a lot of data and involves a lot of code:

- OLTP vs. DSS

This means that the DBA must take an active stance on monitoring and managing Banner's use of memory.

Checking the SGA:

- How often does the database get data from disk vs. memory?
- How often are the stored packages accessed?

Oracle

The main areas the DBA must be concerned with are:

- Data Base Buffers - v\$sysstat
- Shared pool - pins and gets
- Redo logs - Checking the alert logs, v\$sysstat
- Data Dictionary cache - v\$rowcache
- Library cache - v\$llibrarycache
- Sort Areas - temporary tablespace

Storage Management

Banner

Banner databases require active storage management.

- Tables grow constantly
- Indexes become unusable
- Database use causes problems like storage fragmentation and row chaining
- New tables with upgrades

Oracle

The storage management tasks the DBA will be most heavily involved in are:

- Tablespace growth
- Table sizing
- Rollback segment sizing (obsolete with undo space management)
- Restructuring the database
- Resizing/moving tables and indexes
- Tablespace placement
- RAID vs. non-RAID

Disk Storage Management – Banner

RAID

- Important to have for an Oracle system
 - Jbod – Lowest data cost – low overhead
 - 0 – low data cost – load balancing
 - 0, 0+1 – high data cost – most expensive
 - 5 – Highest data cost – safe
 - It takes four I/Os for one write (read data, read parity, write data, write parity)
 - Minimum of three disks, plus 66-87% of space
 - The more disks, the higher the gain
 - Not good with REDO – Sequential vs. random writes

Object Management – Banner

Tablespace division

Banner is installed in one tablespace, usually named DEVELOPMENT. It can be divided into many tablespaces as required by the institution.

- Access reasons
- Performance reasons
- Maintenance reasons
- Multi-institution reasons

Banner Utilities

GURPDED

- SunGard Higher Education-supplied data catalog tool GURPDED
- Run through Job Submission

GURRDDL

- SunGard Higher Education-supplied tool called GURRDDL
 - Used to resize tables
 - Used to resize indexes
 - Handles the referential constraints
 - Handles the permissions
 - Handles the synonyms
 - Can be used to move one or many tables

Tools

These tools are available on an "as-is" basis, subject to a click license. If you require assistance with these Banner Tools, please contact your SunGard Higher Education Account Manager so that SunGard Higher Education Services can provide assistance.

- UI Toolkit

For Banner 7 Forms (based on the Oracle Forms 10g API) -- This toolkit will help clients change the UI of their locally created forms so they look like Banner 7 baseline forms. This is basically a utility written using the Oracle Forms10g API. SunGard Higher Education used this utility on baseline Banner forms to automate some of the User Interface changes in Banner 7.

- API Toolkit

For Banner 7 APIs -- These scripts generate the framework code for the Banner 7 APIs. SunGard Higher Education used these scripts to help create the Banner 7 APIs. These scripts will help clients create their own Banner APIs.

- Automated Test Harness

For Banner 7 APIs (based on UTLPLSQL) -- This utility helped automate some of the testing of Banner 7 APIs. SunGard Higher Education used this utility during Banner 7 development. This utility has been tested only on Windows XP Professional and Windows 2000 operating systems.

It is critical to have the word document created following the guidelines provided in the user guide for this tool. Java Virtual Machine 1.4.2_04 alone has been used.

Oracle Fragmentation

Fragmentation

Databases can fragment, even with the best of software. Fragmentation causes performance problems. Uniform extent size relieves the DBA from worrying about free space fragmentation. Oracle has stated that segment fragmentation does not seem to be a problem until the number of segments grows beyond 1024.

The most common method of defragmentation is export/import.

Before uniform extent sizes, DBAs used SMON to help with free space fragmentation.

Oracle Locally Managed Tablespaces

Methodology

- Oracle has moved to a methodology that reduces table management
- Uses a few basic extent sizes for each tablespace
- Tables are grouped into tablespaces according to size and growth characteristics

Object Management – Banner

General

There are sets of scripts provided with Banner General that are used to compile/recompile database objects.

General Packages, Procedures, Functions, and Triggers
/sghe/banner/general/dbprocs/gendbpr.sql

General Views
/sghe/banner/general/views/genview.sql

Student

There are sets of scripts provided with Banner Student that are used to compile/recompile database objects.

Student Packages, Procedures, Functions, and Triggers
/sghe/banner/student/dbprocs/studbpr.sql

Student Views
/sghe/banner/student/views/stuview.sql

Object management utility

- SunGard Higher Education-supplied tool called GURALTR
- Searches database for invalid objects and builds a recompile script
- Used to recompile database objects (not sys nor system)
 - Packages
 - Package Bodies
 - Triggers
 - Procedures
 - Functions

Exercise 5

How many different Banner code trees does your oraenv script handle?

What is the purpose of the BANNER_LINKS directory?

Further discussion

Even though studbpr.sql is in the Student product directory, what schema owner owns the database procedures in this script?

Compiling Invalid Objects

Steps

- Run utlrp.sql as SYS to compile ALL database objects. This is found in ORACLE_HOME/rdbms/admin
- Run GURUTLRP.sql from the Banner directories to compile all objects. This script must be run as SYS
- Use GURALTR as SYSTEM to recompile ALL non-SYS and non-SYSTEM database objects. This is in BANNER_HOME/general/plus
- BANINST1 owns most database packages, procedures, functions and views. Run GURALTRB as BANINST1 to recompile just BANINST1 objects
- After manipulating ANY database objects, run one of these scripts to fix any broken dependencies
 - Run multiple times to resolve dependencies

Banner Database Triggers

Ownership

BANINST1 does not own the database triggers. Each product owner owns its respective set of database triggers.

Compile scripts for the triggers are located in the product's dbprocs directory in the xxxdbpr.sql file. These must be used in conjunction with GURRDDL when rebuilding tables.

Additional Banner Software

Banner Administrative Software

There is an administrative directory for Banner on the RDBMS server

- \$BANNER_HOME/admin.
 - Banenv (UNIX/NT)
 - Banlogic (VMS)
 - profile.ban
 - upcase.shl

Banner non-database related software

Banner also requires management of all the software outside of the database.

- Executable compiles
- Pro*C
- Pro*COBOL
- SQL*Plus scripts
- SQL*Loader scripts

Banner Pro*C

Introduction

Each product owner may have a set of Pro*C programs, located in /sghe/banner/<product>/c.

- Based on old Oracle RPTs
- Ugly but consistent design

.h – Header Files

.h - Header files (General/c only)

- guarpfe.h - Routines for RPT emulation
- guastdf.h - Routines for C print formatting

.c - C source files

.c - C source files (General/C only)

- guarpfe.c - Routines for RPT emulation
- guastdf.c - Routines for C print formatting
- guaorac.c - Routines for Oracle interface

.pc - Pro*C source files

.pc - Pro*C source files

- gurjobs.pc - Job Submission Pro*C program
- gjrrpts.pc - Banner Reports print program

Compile scripts

The compile scripts for each product's set of Pro*C programs are in the corresponding MISC directory (the same applies for OVMS and NT).

- General Pro*C
/sghe/banner/general/misc/gencmplc.shl
- Student Pro*C
/sghe/banner/student/misc/stucmplc.shl
- Finance Pro*C
/sghe/banner/finance/misc/fincmplc.sql

Banner Pro*COBOL

Introduction

Each product owner may have a set of Pro*COBOL programs, located in `/sghe/banner/<product>/cob`.

`/sghe/banner/general/cob` also contains the Pro*COBOL copylibs.

- `*.pco` - Pro*COBOL source files

Compile scripts

The compile scripts for each product's set of Pro*COBOL programs are in the corresponding product MISC directory.

- General Pro*COBOL
`/sghe/banner/general/misc/gencmpl.shl`
- Student Pro*COBOL
`/sghe/banner/student/misc/stucmpl.shl`
- Finance Pro*COBOL
`/sghe/banner/finance/misc/fincmpl.sql`

Banner SQL*Plus

SQL*Plus scripts

Banner also has SQL*Plus scripts that are used by the various Banner products, located in /sghe/banner/<product>/plus.

General also has some scripts that are used for admin and upgrade purposes.

- LOGIN.SQL
- GURALTRB
- GURRDDL
- GOSTAGE

Banner SQL*Loader

SQL*Loader scripts

Banner also has SQL*Loader scripts that are used to load outside data, located in /sghe/banner/<product>/loader.

- Test scores
- EDI
- *.ctl - SQL*Loader control files

Exercise 6

What SQL will compile all Banner objects?

What objects does BANINST1 not own?

Where will you find the file(s) that set up the environment for each Banner code tree?

Further discussion

What is the fully qualified name of the script to compile the GSASECR.fmb file?

Banner - Access Methods

Considerations

Considerations need to be made for:

- Access to the Banner system
- Database(s)
- Forms
- Report output
- User vs. developer access
- Banner System vs. SQLPLUS
- Microsoft Access
- Financial aid load files

Creating an Banner Database

Introduction

Production databases are usually created sometime after the initial Banner installation by the site.

Memory sizes, block size, disk storage requirements, network considerations, and user load must be taken into account when configuring the production database.

Banner Database creation methods

Creation is done through one of the many tools available.

- Command line method has more creation flexibility
 - Refer to createdb.sql
- GUI mode (DBCA) is easier, but requires modification before the Banner data can be loaded
- Clone the PPRD database

Cloning a Database

Introduction

Cloning is done through a database recovery mechanism.

- Database files are copied to new directory structure
- Produce a control file re-creation script by issuing as sys:
`alter database backup controlfile to trace;`
- New instance is "recovered" using modified control file, and new control files are created

Cloning – Oracle 8i or later

Use RMAN to make a backup of the source database. RMAN preserves the control file entries and DBID.

The clone database is created and opened NOMOUNT.

Run the duplicate command in RMAN to perform the clone.

- The procedure is fully documented in the [Backup and Recovery Concepts](#) manual.

Banner - Backup Strategies

Cold backups

- Involves taking the database offline
- Should be performed occasionally

Online backups

- Allows the database to remain in service
- Requires a large UNDO to 'queue up' the DML
- Must be in archive log mode

RMAN

- Only backs up the changed blocks
- Uses the Large pool

Exercise 7

Why is the method used to clone Oracle7 databases not a good one for Oracle8i and later databases?

Which backup methods require that the database be down?

Can RMAN backup a non-archived database?

Self Service Banner (SSB) Products

Introduction

The Self Service Banner ... products are an add-on set of software functionality that provides a self-service browser interface to many Banner functions.

- Not "Internet Native" Banner
- Customizable look and feel
- Rules-defined access and display
- Implemented by using database packages and procedures
- Generates dynamic HTML pages from the database
- Uses Oracle Application Server's modplsql module
- Uses Oracle's PL/SQL Toolkit package
- Banner has an administrative tool called Web Tailor

Documentation

- The end user guides are available through Banner Bookshelf through the Product User Manuals
- The Upgrade Guides and Release Guides are available through Banner Bookshelf
- Each product has its own set of Guides
- Web Tailor has its own set of Guides

Software installs

Self Service Banner... installs are done on the Banner RDBMS server. Packages are built in the database.

The software is installed in Self Service product directories:

- /sghe/banner/aluweb
- /sghe/banner/facweb
- /sghe/banner/finweb
- /sghe/banner/genweb
- /sghe/banner/payweb
- /sghe/banner/scomweb
- /sghe/banner/stuweb
- /sghe/banner/wtlweb

Software structure

Each Self Service Banner ... product directory contains some of the following structure:

- c - C programs used in web applications
- dbprocs - Web packages
- gif - graphics objects for buttons
- html - Product web pages
- misc - Web administrative scripts

Installation

There is no Banner install on the web server.

The server is configured to connect to the Banner database through Oracle Net Services.

Self Service Forgif and .htm files are copied from the database server to the web server.

- An Apache listener is created to present the initial page for general access to the SSB system, homepage.htm

Web Server Install

From the web server, for each Banner instance:

- Create a DAD (Database Access Descriptor)
- Created with several versions of Oracle's GUI tools
- Uses a generic username/password/connect descriptor that logs into the database on behalf of the SSB user
 - `http://host.domain.edu:1810`

DAD – AS10g

- Go to the AS Welcome page:
 - <http://host.domain.edu:1810>
- Click on HTTP Server
- Click on Administration
- Click on PL/SQL Procedures
- In the DAD section, click on Create
- Click General
 - DAD Name: /pls/SID
 - Username: www2_user
 - Password: u_pick_it
 - Connect String: DBhost:1521:SID
 - Connect String Format: SIDFormat (host:port:SID)
 - Default Page: homepage.htm

SSB Web Server Directories

Copy .gif and .htm files for each product from the directories on the RDBMS server to the appropriate directory on the web server:

- /sghe/banner/*web/gif → /sghe/ssb/SID/*gifs
- /sghe/banner/*web/htm → /sghe/ssb/SID/*help

Web Server Listener

- Configure the Web server to 'serve up' SSB
- Create a virtual host with:
 - Set DocumentRoot to the /u01/app/sghe/ssb/SID directory
 - Set Port to the Port that the users will use
 - Set Listen to the Port that Apache will listen for
 - Set DirectoryIndex to homepage.htm
 - Set "Before Procedure" value to "twbklist.p_main"

homepage.htm

The next step is to set up the homepage.htm file.

- Copy from:
`/u01/app/sghe/ssb/SID/wtlhelp/homepage.htm`
- To:
`/u01/app/sghe/ssb/SID/homepage.htm`
- Replace //your server name:port/test/owa/ with your site's mod_plsql reference and DAD
 - i.e. //host.domain:9030/pls/SID/
 - /pls for modplsql alias
 - /SID for the DAD name

Objects

The only Banner software to manage on the web server are the web objects needed by the Self Service products:

- .gif files
- .htm files
- Web Tailor image files
- Java
- CSS for cascade style sheets

These will be updated as new versions come out.

Modify database

Run the following SQL statements on the database server:

```
UPDATE twgbparm
  SET twgbparm_param_value = 'WWW2_USER'
  WHERE twgbparm_param_name = 'WEBUSER';

UPDATE twgbwrul
  SET twgbwrul_cgibin_dir = '/pls/SID';
```

RDBMS Software

There are sets of scripts provided with Banner that are used to compile/recompile database objects.

Examples

- Stuweb Packages, Procedures, Functions, and Triggers
/sghe/banner/stuweb/dbprocs/bstudbpr.sql
- Genweb Packages, Procedures, Functions, and Triggers
/sghe/banner/genweb/dbprocs/bgendbpr.sql

Client

- Netscape 4.x
- Explorer 5.x/6.x
 - Generally, stay a release or two back from the latest and greatest browsers
- Supports other browsers (Opera, Mozilla, etc.)
- Could use Windows versions from 95 to the present
- Can use MacOS
- SSL or VPN for secure access
- High encryption is not available on older OS versions

Exercise 8

Where is each web page generated from?

What Banner software must be on the Web server?

How does a SSB user connect to the database?

Further discussion

How would you reload the PL/SQL toolkit in the database?

Internet Native Banner - Parts

User Interface

In Internet Native Banner, the Oracle forms software has been modified to run in two parts:

The forms runtime process runs on an application server as either:

- Forms CGI (ifcgi)
- Forms Servlet (ifsrv)
- A Java applet that performs the display and data entry tasks on the client (Jinitiator)

The forms are "tagged up" with HTML tags and presented to the applet running in the browser in standard http 1.1 protocol.

Internet Native Banner - Installs

Forms and Reports

- AS10g Forms and Reports Services Standalone
- Forms and Reports is included with OAS10g Enterprise Edition release

In any case, always apply the latest supported Oracle Forms and Reports patch sets.

Preparing to generate forms

- Copy .fmb, .pll, .mmb, files from database server to /u01/app/sghe/inb/*SID*/forms/fmb
- Copy xxxform.shl files from database server to /u01/app/sghe/inb/*SID*/forms/bin
- On the INB server, copy \$ORACLE_HOME/bin/fgenm.sh to /u01/app/sghe/inb/*SID*/forms/bin and edit the copy appropriately
- Edit \$ORACLE_HOME/network/admin/tnsnames.ora file for appropriate *SID*
- Edit xxxform.shl files - change *password* to *password@SID*

Generating forms

```
cd /u01/app/sghe/inb/SID/forms/fmb
. /u01/app/sghe/inb/SID/forms/bin/fgenm.sh
sh /u01/app/sghe/inb/SID/forms/bin/xxxform.shl
tar cvf fmx.tar *.fmx *.plx *.mmx
mv fmx.tar ../fmx
cd ../fmx
tar xvf fmx.tar
```

Helpful additions

- Create /u01/app/sghe/inb/SID/forms/bin/allform.shl script
 - Invokes fgenm.sh
 - Invokes all the xxxform.shl scripts
- Copy genform.shl script to oneform.shl script
 - Modify to look for all necessary .pll and .mmb files
 - Invokes fgenm.sh
 - Uses a parameter to generate one form only

httpd.conf

- Configures the Apache Web server to 'web up' INB
- Edit the HTTP listener via the EM Console
 - Set DocumentRoot to the top level directory for your server
 - Set Port to the Port that the users will use
 - Set Listen to the Port that Apache will listen for
- Another option is to create an include file to set up a virtual host for access
- Either method requires a restart of the HTTP server

formsweb.cfg

- Servlet configuration file
- Provides several parameters to the browser
 - System
 - User
 - Html page
 - Applet
 - Jinitiator

Jinitiator

- Different versions of Jinitiator can be installed and used simultaneously
- Users download Jinitiator once
- Subsequent uses of Internet Native Banner use the downloaded copy on the client machine
- Upgrades to new versions of Jinitiator will require new downloads

Forms and Reports environments

- Copy \$ORACLE_HOME/forms/server/default.env to \$ORACLE_HOME/forms/server/*sid*.env

- Add the following lines via the EM Console

```
TWO_TASK=SID  
TNS_ADMIN=/u01/app/sghe/inb/SID/conf
```

- Change the following lines for your environment

```
FORMS_PATH=/u01/sghe/inb/SID/forms/fmx  
REPORTS_PATH=/u01/sghe/inb/SID/reports/rep  
GRAPHICS_PATH=/u01/sghe/inb/SID/forms/fmx
```

Web based output

- The INB Job Submission "save and print" temporary files are located on the web server
 - They utilize the DAD technology to be presented over the web
 - Form GJIREVO handles the program and report output files
- **They require a DAD (Database Access Descriptor).**
 - Reference the DAD created by example in the workbook
 - Enter the values on GUAUPRF

- **For "Enter the name of your Web Output URL," enter**

```
http://host.domain:port/pls/<SID>
```

.gif files

The database server contains .gif files for Internet Native Banner Forms in the form of Java Archives.

- On your database server change directory to the location of your jar files
\$BANNER_HOME/general/java
- Copy these to the /u01/app/sghe/inb/java directory on the INB server

Files in general/extras

fmrpcweb.res (copy or ASCII FTP to \$ORACLE_HOME/forms and rename to fmrweb.res)

For F1 display help, add this line to fmrweb.res

```
112 : 0 : "F1" : 1004 : "Help (Item Properties)"
```

Starting INB

Starting INB – Windows

- Click on Start -> Settings -> Control Panel -> Services
- Stop/Start the OracleHTTPServer service
- Stop/Start the OracleReportServer service
- Execute INB (Internet Native Banner)
- Start your browser with the URL (example only):
- `http://host.domain:port`
- When the GUAINIT form starts, enter

Username: saiusur

Password: u_pick_it

Starting INB - UNIX

- Log on to the web server as oracle
- Set up environment:
 `. oraenv` (use oratab entry for AS10g)

or
 `source $ORACLE_HOME/bin/iasenv.sh`

 `$ORACLE_HOME/opmn/bin/opmnctl startall`
- Start a web browser and enter your URL:
 `http://host.domain:port`

Exercise 9

What is the major difference between base.htm, baseie.htm or basejini.htm?

You just installed a new patch to Forms – what do you need to change for a new version of Jinitiator?

Upgrades



Introduction

This section discusses upgrading Banner.

Upgrading Banner

Introduction

The process of maintaining Banner involves frequent upgrades for both enhancement and error correction purposes. These tasks will involve using:

- The Customer Support Center
- Patches
- Upgrades

Customer Support Center

Introduction

The Customer Support Center is SunGard Higher Education's web site for:

- Information on the Banner products
- FAQs
- Finding information on defects
- Finding and downloading patches
- Entering technical request contacts

You must be a valid SunGard Higher Education customer to access the Customer Support Center.

Account

- Requires setting up an account
- Specific username and password
- To set up an account:
 - Start at <http://www.sungardhe.com>
 - After logging into the account, the Customer Support Center home page displays several options.

Support

- The Customer Support Center is the avenue through which to report Banner problems
- ActionLine support is now available via the Customer Support Center
- Customer Support Center goes into the same helpdesk pool as the ActionLine

Banner Patches and Upgrades

Banner patches and upgrades

Most enhancements and patches will be downloaded from the Customer Support Center. As issues are resolved and error corrections developed, these fixes will be bundled into patch sets and posted on the Customer Support Center electronic download page.

- These patch sets are compressed and encrypted in a .trz file
- Must use a special program to decrypt it.
 - Windows – edunld.exe
 - OpenVMS and UNIX – download decryption tool

Finding patches

The first step is to use the Known Issues search engine to locate defect and associated patch numbers. New patch postings are also broadcast on the SunGard Higher Education listserv BPOST.

Once the proper patches are located, invoke the download gadget to retrieve them.

Downloading patches

There is a web gadget that allows for easy packaging and downloading of patches and the decryption program.

- Follow the directions on the main Customer Support Center page for downloading patches to set it up
- Set up a directory structure to download and decompress/decrypt these patches

Installing patches

Once the patch is downloaded, decompressed, and decrypted, review the install instructions.

Most downloads involve new software or database objects, which will need to be

- migrated to proper software directory
- compiled as needed
- installed into the database as needed.

Managing patches

Patch management is one of the most challenging of the Banner maintenance duties. A methodology and conventions for storing, testing, and applying patches should be developed. Patch installation does not necessarily coincide with upgrade installations.

Banner upgrades

Banner upgrades come in two types and listed as full upgrades:

- Cumulative
- Interim

Upgrades are applied in two parts:

- Upgrade applied to the database
- Upgrade applied to the software

Upgrade dependencies

Each Banner product has its own upgrade procedure.

- These upgrades must be applied in a particular order to build dependencies properly
- There is a dependency matrix for all upgrades in the Banner general FAQ section
- Read Page Six of the [Upgrade Guide](#) for the overview

Database upgrades

GOSTAGE is a PL/SQL program that builds a database upgrade script based on two Banner upgrade tables:

- GUBSMOD - Holds Modification Identifier
- GURSSQL - Holds Modification SQL

Creates and runs a file called DOMOD.SQL which will apply all of the upgrade changes to the database

Software upgrades

The upgrade to the software happens in sync with the database upgrade.

All changes to source code are migrated to the software tree during this stage.

All Pro*C, Pro*COBOL, and Forms executables are compiled and moved to the appropriate executable trees.

Forms source (.fmb, .pll, .mmb) need to be migrated to the INB web server(s).

Managing upgrades

- How many upgrades - the bare minimum
 - Remember the O/S and Oracle
 - The database server is not the only upgrade
- Timing the upgrades
 - READ THE UPGRADE DOCUMENTS
 - Media verification
 - Always apply upgrade to the baseline database SEED first

Step dependencies

- Page 8 of the Upgrade Guide contains a chart that will help you through the upgrade process
 - The chart lists which steps will be done and which are not applicable
 - The last column is to be used to get a time value for that step
- Once you have done an upgrade first in a non-production instance, you then know how long the upgrade will take
 - Users are always asking "How long will it be down?"

Security



Introduction

This section discusses managing security in Banner.

Security Overview

Authentication vs. Authorization

With access to Banner databases now possible from the Internet, security had to be redesigned to handle the following two issues:

- Authentication
How do I know you are who you say you are?
- Authorization
Can you do this?

Oracle Security

- Privileges
 - Object Privileges
 - System Privileges
- Roles
 - Default Roles and passwords

Banner Security

- Objects
- Classes
- Roles
- Users
- BANSECR
- Security Routines
- Value based security

Oracle Access Rights

Access levels

Oracle provides two levels of access to database objects and data:

- Object level privileges
 - Specific access to specific objects
 - Example: SELECT ON SPRIDEN
- System level privileges
 - General access privileges to groups of objects
 - Select on any table

Oracle Roles

Introduction

- Oracle roles are pseudo-users
 - Many system and/or object privileges are granted to the role
 - These roles are then granted to Oracle users
- Roles provide a grouping ability
- Roles can be nested

Which Privilege?

When there are many access paths to an object, the most powerful of the underlying privileges will be in effect.

Roles can be passworded, requiring knowledge of the password to invoke the role.

Banner Security Model

Role-based security

Banner takes a novel approach to security, using roles with passwords.

The end user does not know the password to invoke the role; the process the user is trying to perform has the ability to invoke the role. Once the process ends, the role is revoked.

Banner Security Classes

Banner processes are grouped into classes. Classes are a sort of pseudo-role.

- End user is granted access to the class
- End user is granted an associated role without knowledge of the password
 - BAN_DEFAULT_Q - Query-only use
 - BAN_DEFAULT_M - Maintenance use
- End user is also assigned a default role

Banner Security Process

- When the end user starts the process, access to the class is checked
- If access to the class is granted, the associated role is invoked
- End user has temporary access to the database objects
- End user has access rights revoked at process end

Banner Security – GSASECR

GSASECR

GSASECR is a special Banner form designed to manage Banner security.

GSASECR should not be available to anyone but a security administrator. BANSECR is the only allowed user of the GSASECR form.

Purpose

GSASECR is used to manage:

- Objects (Processes/Jobs, Forms)
- Classes
- Roles
- Users
- System Security

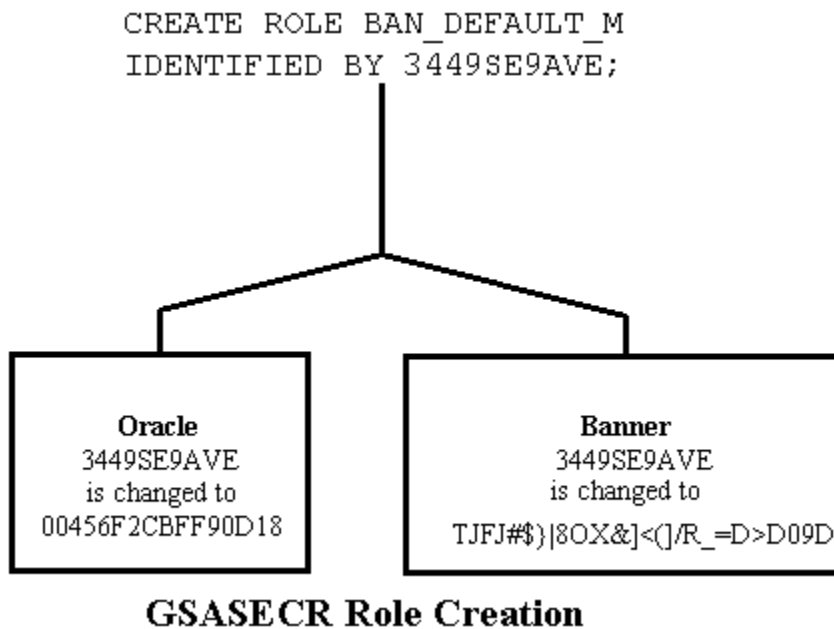
Banner Security – GSAVPDI

GSAVPDI

- Functions the same way as GSASECR
- Must be logged in as BANSECR
- On first entry, you select one of the following:
 - User maintenance
You set up what institution code the user will use
 - Institution code maintenance
Identifies all valid institution codes

Banner Role-Level Security

Diagram



Banner - Adding Security to Site Processes

Components

- The SQL code for security
- Installing security in your programs
- C programs
- COBOL programs
- Forms

Custom Code

Definition

- All objects created that are not baseline are required to be added to the Banner system
 - GUAOBS
 - GSASECR
 - Object maintenance
 - Class maintenance
 - User maintenance
 - Job Submission

Code Invoking Process

- The object name is passed to the procedure
- The seed numbers are passed to the procedure
- The three-phase decryption takes place
- The role password is invoked
- The local variables are cleared

SQL – FAQ#1979

Supporting files

Each SQL requires a supporting file to submit it. This file needs to be in the correct path for its OS:

- VMS -- in logical GEN\$COM or BAN_COMS
- UNIX -- in environment variable \$BANNER_LINKS or \$PATH
- WinNT/2000 -- in environment variable PERL5LIB or
%BANNER_HOME%\general\misc

VMS

```
VMS -- test_sql.com
-----
$! Test of Job Sub process to run SQL SCRIPT.
$ sqlplus 'UIPW' @test_sql
```

UNIX

```
UNIX -- test_sql.shl
-----
# test_sql.shl
# mtytler 7/31/97 testing job sub with sqlplus
sqlplus $UIPW @test_sql.sql
```

Windows

```
WinNT/2000 -- test_sql.pl
-----
# test_sql.pl
# mtytler 4/9/2002 testing job sub with sqlplus
use sctban;
&sctban_determine_os;
&sctban_os_specific_env;
&sctban_jsub_env;
$a1 = $ARGV[0];
$a2 = $ARGV[1];
$a3 = $ARGV[2];
$a4 = $ARGV[3];
$a5 = $ARGV[4];
$banner_exe = $ENV{"BANNER_EXE"};
open(PPROCESS,"|sqlplus ${sctban_user_pass} \@test_sql.sql");
close(PPROCESS);
```

SQL – Actual file

test_sql.sql

```
SQL File -> test_sql.sql
-----
rem mtytler 4/9/2002
rem test_sql.sql
rem SQL SCRIPT TO TEST JOBS SUB PROCESS
spool test_sql;
select * from dual;
spool off;
```

- test_sql.sql needs to be in the correct path for its OS
 - VMS -- in logical ORA_PATH
 - UNIX -- in environment variable \$ORACLE_PATH
 - Windows -- in environment variable SQLPATH
- Finally, add the object in GSASECR, GUAOBS and job submission

C programs – FAQ#2587

Process

- Easiest of the processes to implement
- The makefile includes the necessary object file at link time after the program is compiled
- Insert the following code, compile it with the provided sctproc(.mk, .com, .pl), and then add the object in GSASECR and GJAJOB

```
#include "guarpfe.h" /* Holds rptopen function */

EXEC SQL INCLUDE guaorac.c;
/* Holds the user_pass variable below */

int main(int argc,char *argv[])
{
    rptopen(user_pass,argc,argv);
    /* rptopen does the proper setup for security.*/
    login();/* Calls Banner Security from guaorac2.pc */
}
```

Required change

If your program is a copy of baseline
(i.e. - \$BANNER_HOME/general/c/gjrrpts copied to gzrrpts.pc), then you must change a
line in the program before recompiling it.

Find the line

```
strcpy(rpt_name, "GJRRPTS");
```

and change it to

```
strcpy(rpt_name, "GZRRPTS");
```

COBOL Programs

Steps

Insert the line

```
SQL EXEC INCLUDE SETSEED END_EXEC
```

into the declarative section.

Move the program name into OBJECT-NAME .

Insert the line

```
SQL EXEC INCLUDE SETROLE END_EXEC
```

into the PROCEDURE DIVISION.

Refer to the COBOL standards section in General_trm.pdf file (section 1-49).

Forms

Steps

Attach the library GOQRPLS.

Create the LOAD_CURRENT_RELEASE trigger.

Reference the G\$_REVOKE_ACCESS and the G\$_VERIFY_ACCESS triggers from the GOQOLIB reference library form.

If you attach the new G\$_FORM_CLASS, G\$_APPL_FORM_CLASS, G\$_INQ_FORM_CLASS, or G\$_VAL_FORM_CLASS to your form, you will automatically inherit these triggers.

Notes

If you do not reference those triggers from GOQOLIB, you will need to create a pre form trigger and a post form trigger to accomplish the same effect.

Refer to the Security manual, section 1-11.

If you have created a custom form (for example, if you copied SPAIDEN.FMB to form WPAIDEN.FMB), copy SOQMENU.FMB to WOQMENU.FMB and place it in the FORMS_PATH.

Reports – FAQ #1582

Steps

You need to create the security check sums for the new report TESTRPT.REP:

```
gurchks bansecr/u_pick_it@database testrpt.rep 1.0 plus33 G  
BAN_DEFAULT_M
```

This should create a security check sum value.

Enter the report in GSASECR and GJAJOBS.

The user also needs permission to the underlying tables – FAQ#1729.

Self Service Banner

Roles

The Self Service products are secured by a valid PIDM.

Each web product requires a role to be set up:

- STUDENT role = if ID has a SGBSTDN record
- FACULTY role = if ID has a SIBINST record
- EMPLOYEE role = if ID has a PEBEMPL record
- ALUMNI role = if ID has an APBCONS record
- FINANCE role = if ID has a FOMPROC record
- WTAILORADMIN role = assigned (TWGRROLE)

Procedural call

To create your own package within the SSB products, a procedural call must be added. It will validate that the user has a valid PIDM.

```
PROCEDURE twbkwbis.F_ValidUser(pidm number)
```

Example

```
CREATE OR REPLACE PACKAGE BODY Hello_Chris
IS
PROCEDURE P_DisplayHello IS
pidm number;
BEGIN
IF NOT twbkwbis.F_ValidUser(pidm) THEN
RETURN;
ELSE http.print('Your internal id number is :'|pidm);
END IF;
END;
```

Adding Processes to Banner Security

Steps

After the site process has had the security code added to it, the object must be added to the Banner Security tables.

The steps are:

- Create the new object in GSASECR
- Add the object to Banner Classes
- Assign a role to the object/class combination
- Resynchronize the class/user grants
- Don't forget GUAOBS!

BANSECR - Tables and Objects

Components

- Nine tables
- Two views
- One sequence
- Four database packages
- Three synonyms

BANSECR tables (9)

- GUBIPRF - Site profile record
- GTVCLAS - class titles and last_sync date
- GURAOBJ - all valid Banner objects, version numbers, default role to invoke
- GURUOBJ - all objects given directly to a user (not through a class)
- GTVVPDI – Contains the institution codes for your system
- GURUCLS - associates Oracle UserID with Banner classes
- GURALOG - security validation log (must be TRUNCATED on occasion as it grows)
- GUBROLE - encrypted passwords for the Banner roles
- GURSOLL - history of all SQL issued by the GSASECR form

GURSQL – AUDIT BANSECR

Besides auditing any sql statement, you can audit the BANSECR users.

Create site-specific BANSECR users (Security manual 1-9):

```
define system_password=manager
define sys_password=change_on_install
define bansecr_password=secret1
define baninst1_password=secret2
define u=bansecr_example
define u_password=u_pick_it
define temp_tablespace=temp
define ur=usr_gsasecr_bansecr_example
start gssacct (also runs gsspriv, gssbasg,gsssysg)
start gssusrm
start gssusra
start gssroilm
```

This way, you can also audit the security officers.

BANSECR views

- GUVRPRV - table permissions given to a role
- GUVDFTR - to show a user's default role

BANSECR sequence

- GUBOSEQ - One-up number used to sequence the records in the GURSSQL (SQL history) table

BANSECR database packages

- G\$_SECURITY_PKG - procedures to verify an end-user and their access to perform object authentication
- G\$_AUTHORIZATION_PKG - security routines used for job submission, and routines to synchronize the Oracle grants with the Banner class definitions
- G\$_VDPI_SECURITY - VPD to define the home context for multiple institutions
- G\$_OREP_SECR_PKG - Generated in support of the SunGard Higher Education security patch

BANSECR synonyms (3)

- BANINST1_SQL_PKG - points to a BANINST1 owned package that is granted ONLY to BANSECR
- G\$_SECURITY - public synonym for the G\$_SECURITY_PKG
- G\$_CHK_AUTH - public synonym for the G\$_AUTHENTICATION_PKG

Encryption

Process

Banner uses a three-phase encryption process for authentication purposes. For databases that are available on the Internet, this serves to validate that processes accessing the database are the proper processes.

Seed encryption numbers

Seed numbers are set in the database and the processes. When the process is invoked, the database and the process exchange these seed numbers to verify authenticity. If the seed numbers do not match, the role is not invoked.

Seed numbers

- Database seed numbers are changed first
- Recompilation of all forms and programs must be done after the seed numbers are changed
- This may also have to be done if an upgrade modified any of the objects listed below.

SEED number objects

Seed numbers are set in:

- The form GSASECR in Profile Maintenance
- Pro*C - guassed.h file
- Pro*COBOL - guasetr.pco file
- Forms - goqolib.fmb - g\$_verify_access trigger

Under G\$_VERIFY_ACCESS, there are red arrows on four sub-items (G\$_APPL_FORM_CLASS, G\$_INQ_FORM_CLASS, G\$_VAL_FORM_CLASS, MENU_TYPE). If you change the seed numbers in G\$_FORM_CLASS, the changes should cascade through the other four. Check to be sure that all five items have been changed.

- Graphics - goggrph.ogd

Under G\$_VERIFY_ACCESS, there are red arrows on four sub-items (G\$_APPL_FORM_CLASS, G\$_INQ_FORM_CLASS, G\$_VAL_FORM_CLASS, MENU_TYPE). If you change the seed numbers in G\$_FORM_CLASS, the changes should cascade through the other four. Check to be sure that all five items have been changed.

- Batchsecurity.java

Banner 8.0 Security Enhancements

Common Security Enhancements

Banner 8.0 contains a number of changes to how Banner handles security and auditing. The focus of these changes is on the following improvements:

- Greater efficiency in security maintenance
- Enhanced Banner user account rules
- Additional user login controls
- Improved security auditing capabilities
- Secured delegation of responsibilities for distributed users

PIN Security Enhancements

PIN enhancements have been made to Banner 8.0 that perform the following functions:

- Enhance the characteristics of PINs to meet industry standards for strong passwords
- Store PINs in the database as a cryptographic hash
- Provide a PIN reset mechanism that is based on providing answers to questions specific to the individual user

Enhanced characteristics of PINs

Going forward, Banner 8 will:

- allow minimum and maximum PIN lengths to be configurable by an institution
- allow institutions to designate whether or not PINS must include alpha and/or numeric characters
- allow institutions to restrict reuse of PINs, if desired.

For more information about these and other Banner 8.0 enhancements, please refer to the *Banner 8 General Release Guide* or the Education Practices *Banner 8 Common Conventions* workbook.

Job Submission



Introduction

This section discusses Job Submission in Banner.

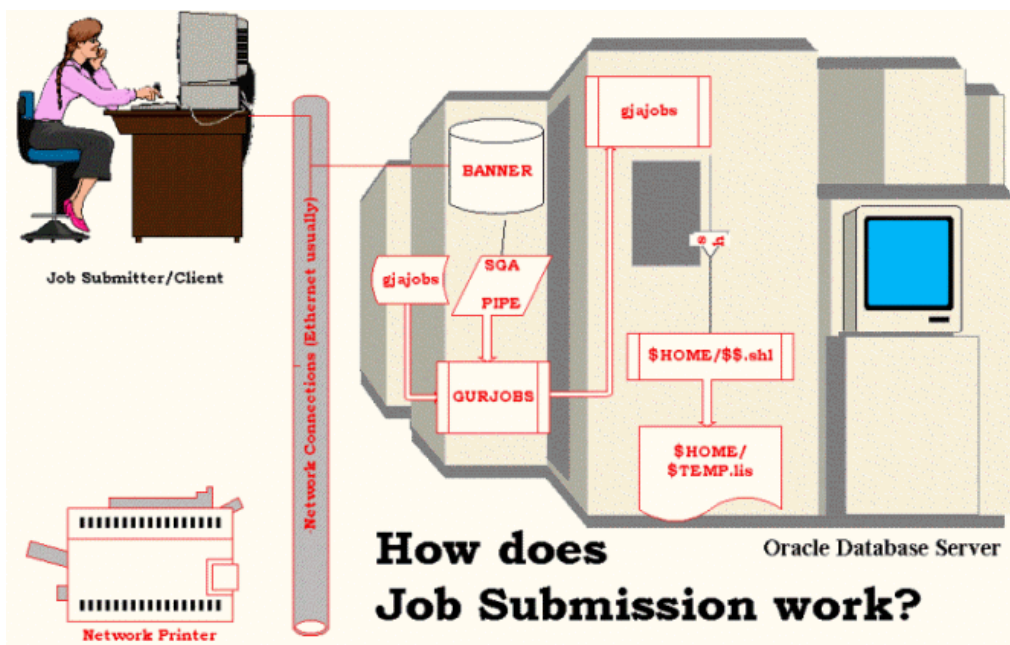
Job Submission Overview

GJAPCTL

Job Submission is a multi-step process:

- User requests process in GJAPCTL Banner form
- The form passes request through the package *dbms_pipe* to a Pro*C program running on the RDBMS server
- GURJOBS

Diagram



GURJOBS

- Starts job submission running in the background on the RDBMS server

GJAJOBS

- The program on the RDBMS server then builds a script to execute the requested job and passes that to the Operating System
- Requested program executes

Results are then:

- Stored in the database
- Stored on a file system
- Printed

Process Types

There are four types of jobs that are currently supported by Banner:

- C - Pro*C
- E - Pro*COBOL (P when executed via a shell script)
- P – Procedures (scripts)
- R - Oracle Reports

Process, Parameter, Printer Definition

Process definition

Processes must be defined to Banner before they can be run.

There are several Banner forms used to define processes to Job Submission:

- GJAJOB - Job Definition
- GJAPDEF - Parameter Definition
- GJAPVAL - Parameter Values
- GJAPDFT - Parameter Default Values

Log into Banner to view the forms.

Parameter definition

These parameters must be accounted for in the Job Submission parameter handling routines of the process.

Defaults can be set in Banner and/or in the program logic.

Printer definition

Printers must be defined to the O/S.

GTVPRNT is a separate form for configuring the print handling. GTVPRNT must be populated before printers can be referenced.

- Define Printer port
- Define landscape printer codes
- Define portrait printer codes

Job Submission - The Process

Definition

- This process is different on every platform
- Operating System and Network requirements may make modification of Job Submission processes mandatory
- Job Submission uses the GUBINST table to determine OS specifics that must be built into the job stream. (GUAINST form displays GUBINST data)

User view

- Job Submission begins in the GJAPCTL form
- Process is validated
- Process parameters are inserted into the process run parameters table GJBPRUN using the GJBPSEQ sequence
- GJAPCTL then performs a call form to GUQINTF, the Job Submission interface form

Technical view

- GUQINTF determines that request came from Job Submission
- GUQINTF fires JS_HOST_COMMANDS trigger to build a command string to pass to GURJOBS
- Contains:

Job Type	Printer name
User name	Special forms
Password	Submit time (OpenVMS only)
- GJBSEQ sequence
- GUQINTF then fires the PIPEIT form level trigger
- PIPEIT uses DBMS_PIPE.SEND_MESSAGE to send the command to GURJOBS, executing on the RDBMS server
- GJAPCTL status line is updated, showing Job Submission name and sequence number
- GURJOBS initiates GJAJOBS.SHL file, using the SYSTEM function to start it

GJAJOBS Parameters

GJAJOBS.SHL reads the parameters passed to it by GURJOBS and builds a temporary shell file to execute it.

Parameters:

#1 is the process name	#5 is the one up number
#2 is the process type	#6 is the printer name
#3 is the user ID	#7 is the form name
#4 is the password	#8 is the submit time

Process execution

- TEMP.SHL file executes where the process picks up the program parameters from GJBPRUN based on the sequence number
- GJAJOBS.SHL determines from the PRNT variable if output is to be stored in the database
- If it is, then the GURINSO process is invoked to insert the data into the database
- Otherwise, DEFAULT_PRINTER and LOCAL_DIRECTORY determine output placement in the form GJAJPRF

Windows – SCTBAN.PM

- Perl script used with printing
- Read before printing starts
- Can set up individual printers

```
if ($sctban_osname eq "WNT")
{
  if (uc($sctban_form) eq "HPP17")
  {
    open(PRNT,"|nt_print ${sctban_printer_name} HPP17.frm
    ${sctban_file_name}.lis > temp.lst") ;
  }
}
```

The HPP17.frm is a printer specific code file (HP – PCL)

Output Destination

When submitting the job, the user can specify the destination. If the destination is not defined, the default printer for the job is used. DATABASE option can be specified to have the output placed in the database.

- GUBOUTP
- GUROUTP

Reviewing the Output Online

- After the output is placed in the database, it can be reviewed using the GJIREVO form
- User then has the option of saving the file locally or printing it
- User is also "supposed" to delete it when finished

The *jobsub* User

Purpose

- A UNIX user ID
- Part of the banner user group
- Used to start/stop the GURJOBS process
- A good idea is to have a separate sub-directory for each database

GURJOBS Startup and Shutdown

GURJOBS

The GURJOBS process must be run from the OS level. It is submitted to UNIX via the `nohup` command:

```
nohup sh $BANNER_LINKS/gurjobs.shl  
        userid/password >jobsSID.log 2>&1 &
```

GURSTOP.SQL is submitted from SQL*PLUS to stop GURJOBS.

On Windows machines, it can be run in two ways:

- As a service
 - Easy for anyone to start
 - Uses a local user with a parameter of the batch job
 - Refer to the workbook and FAQ#2018
- As a batch job
 - Must set environments to be sure it is being run in the correct database
 - Then run the job
 - Refer to FAQ#2030

Job Submission - Startup and Shutdown

start_gurjobs

```
start_gurjobs
ORAENV_ASK=NO; export ORAENV_ASK
ORACLE_SID=$1; export ORACLE_SID
PATH=/usr/local/bin:$PATH; export PATH
. /usr/local/bin/oraenv
#
nohup $BANNER_LINKS/gurjobs.shl SEED GURJOBS >
start${ORACLE_SID}job.log 2>&1 &
```

To start the GURJOBS process for the SEED database:
sh start_gurjobs SEED

Stopping GURJOBS

The GURJOBS process must be stopped from SQLPLUS. It is submitted to SQLPLUS through a Unix script.

It is a SQLPLUS script:
GURSTOP.SQL

GURJOBS automatically stops after four days with no activity.

stop_gurjobs

```
stop_gurjobs
ORAENV_ASK=NO; export ORAENV_ASK
ORACLE_SID=$1; export ORACLE_SID
PATH=/usr/local/bin:$PATH; export PATH
. /usr/local/bin/oraenv
#
sqlplus -s genlprd/u_pick_it @gurstop> stop${ORACLE_SID}job.log 2>&1
```

To run the stop_gurjobs script for the SEED database
sh stop_gurjobs SEED

Job Submission – Cleanup

Cleanup

- One must set up scripts to automatically cleanup the various jobsub directories
- If centralized, output cleanup is easier
- If output is routed to the user's directories, cleanup is more difficult
- Must also clean up the database tables
 - GUBOUTP
 - GUROUTP

Summary/Review Objectives

Summary

- Resize tables
- Create a production database
- Configure a database
- Export and import data
- Create a backup strategy
- Locate Banner source code
- Apply a Banner upgrade
- Apply Banner security to users and site-created source code