

**ASUG**  
CONFERENCE

**SAP Analytics &  
BusinessObjects**

Austin, TX • August 31 - September 2

**Dwayne Hoffpauir**

*BI Platform Security for Mere Mortals*

Session #2936



@ASUG365 #SABOUC



# BI Platform Security for Mere Mortals

- About Me

- Employed by EDS / Hewlett-Packard since 1985.

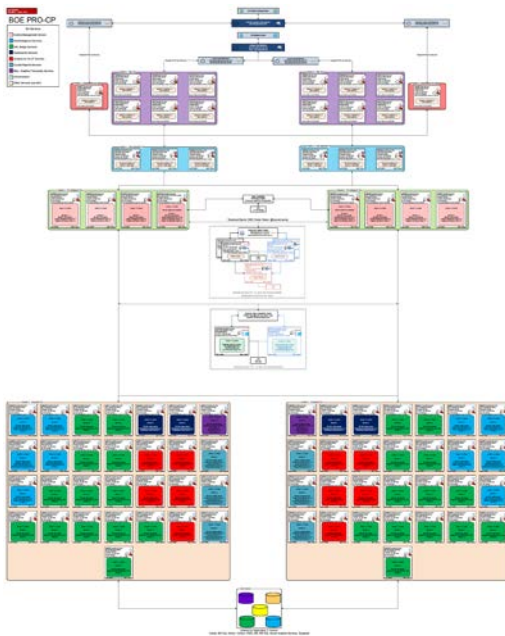


- Representing HP as a customer, not a partner.
- Using Business Objects since 2002 (version 5).
- Business Objects focus is end user (universes, reports), not platform (servers, network).
- Moderator on BOB (Business Objects Board):  
<http://busobj.forumtopics.com>



# BI Platform Security for Mere Mortals

- About the HP Business Objects Platform
  - Running BI 4.1 SP04: Web Intelligence, Crystal for Enterprise, Dashboards, Analysis for OLAP.
  - Approximately 44,000 users and growing.

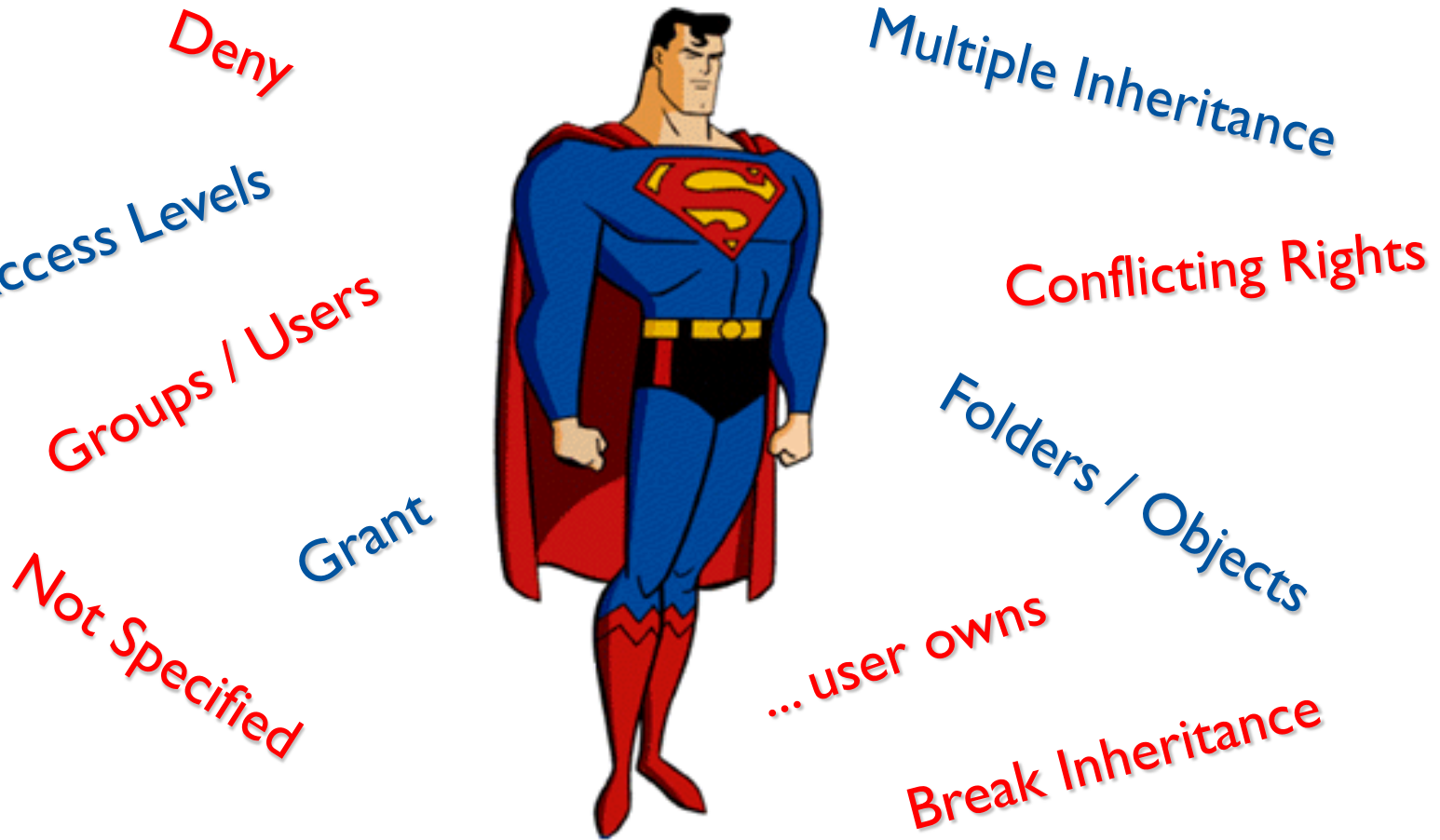


	Servers	CPUs	RAM (gb)
External Apache Web Tier (DMZ)	2	4	8
Internal Apache Web Tier	12	96	384
Java Web Application Tier	6	144	564
BOBJ Intelligence Tier (CMS, IFR, OFR, 1.5 tb SAN)	8	192	752
BOBJ Processing Tier (WebI, Crystal, Dashboards, OLAP)	58	1,392	5,452
Oracle RAC Metro Cluster (2 tb SAN)	8	128	1,024
Windows High Availability Cluster (500 mb SAN)	4	96	1,520
	98	2,052	9,704



# BI Platform Security for Mere Mortals

Ever feel like security requires a superhero?



# BI Platform Security for Mere Mortals

Adopt a “building blocks” approach that mere mortals can implement and maintain.



@ASUG365 #SABOUC

ASUG

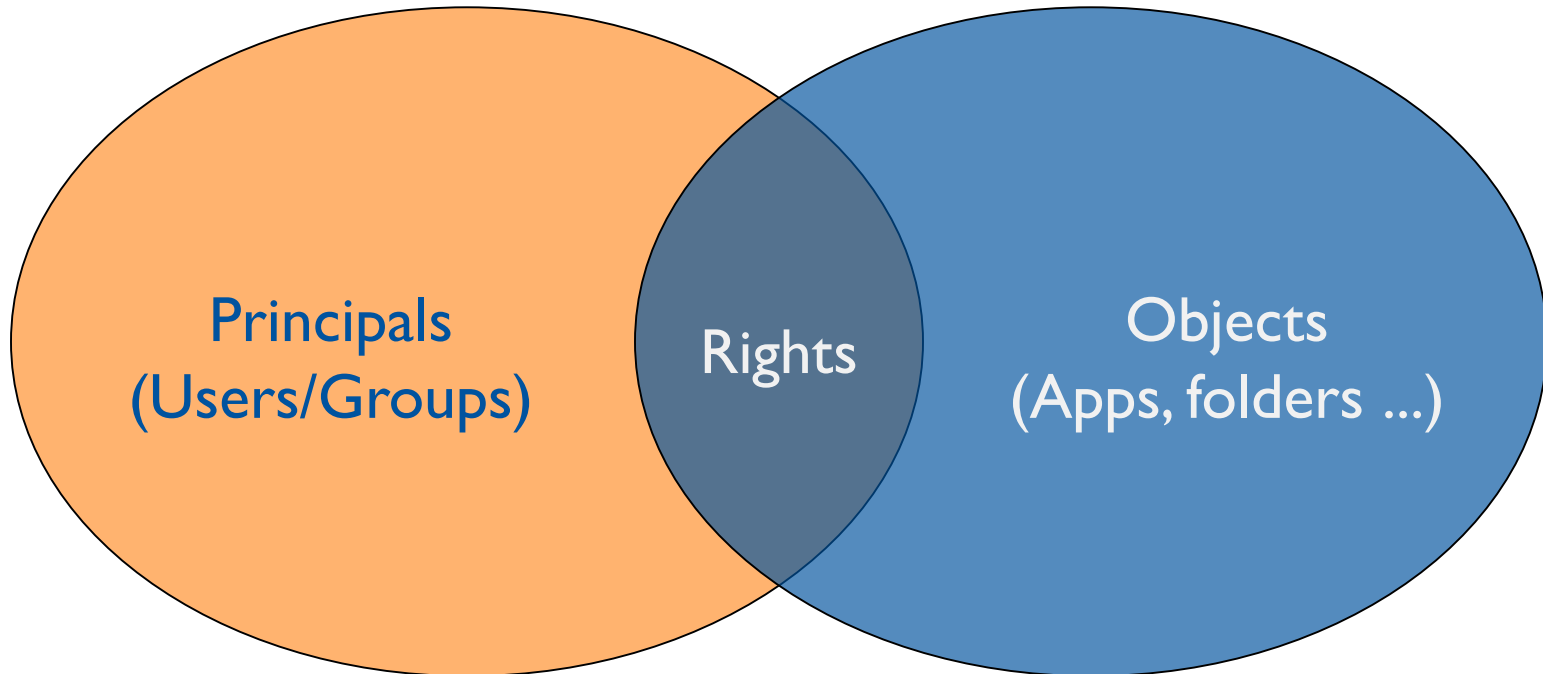
# Learning Points

- **BI Platform Security Concepts**
  - Basic Structure
  - Multiple Inheritance
  - Conflicting Rights
  - Access Levels
- **Building Blocks Model**
  - Guiding principles: aka, things to (try to) avoid
  - Create building blocks: aka, things TO do
  - Create security matrix
- **Tools**
  - Other useful building blocks
  - Central Management Console (CMC)
  - Other resources
- **Recap / Q&A**



# BI Platform Security Concepts

- Basic Structure (Who, What, Where)
  - Principals (who) are given Rights (what) to Objects (where).



# BI Platform Security Concepts

- Principals (Users/Groups)
  - Individual users are assigned to one or more groups.
    - How conflicting rights are handled will be covered later.
  - Parent groups can contain child sub-groups, and child sub-groups can belong to more than one parent group.
    - This “acyclic graph” model (multiple parents) is powerful, but complex.
  - By default, child sub-groups inherit rights from parent groups.
    - It is also possible to “break inheritance,” but should be avoided.
  - By default, users inherit rights from the group(s) to which they belong.
    - Setting user-specific rights complicates maintenance and is another version of “breaking inheritance” that should be avoided.





# BI Platform Security Concepts

- Objects (applications, folders, documents, universes, ...)
  - A key concept is that virtually EVERYTHING is an object.
    - Including “administrative” objects (users, groups, applications, servers, categories, profiles, inboxes, calendars, events, etc.).
  - Objects include documents and universes, but can be Office documents, web links, program objects as well.
  - Objects are stored in folders, but **ONLY ONE** folder.
  - Parent folders can have child sub-folders, but child sub-folders can have **ONLY ONE** parent.
  - By default, child sub-folders inherit rights from parent folder.
    - It is also possible to “break inheritance,” but should be avoided.
  - By default, objects inherit rights from their folder.
    - Setting object-specific rights complicates maintenance and is another version of “breaking inheritance” that should be avoided.



# BI Platform Security Concepts

- Multiple Inheritance
  - Implicit in previous discussions, a given user can inherit rights from multiple places.
  - Groups:
    - User can be in multiple groups.
    - Sub-groups inherit rights from parent groups.
    - Sub-groups can have multiple parent groups.
  - Folders:
    - Sub-folder inherits rights from parent folder.
  - Resolving conflicts from multiple inheritance is discussed on the next page.



# BI Platform Security Concepts

- Conflicting Rights

- Recap:

- Principals are given rights to objects.
- Objects include “administrative” objects and applications.

- Every right has three possibilities:

- Explicitly denied: Always takes precedence.
- Explicitly granted: Applies when otherwise not explicitly denied.
- Unspecified: Not explicitly granted or denied ... considered denied.

- Therefore, conflicting rights are resolved as follows:

- Unspecified + Explicitly denied = Denied
- Unspecified + Explicitly granted = Granted
- Explicitly granted + Explicitly denied = Denied

- Seems explicitly denied and unspecified are the same, right?

- Would seem so, but unspecified is MUCH more flexible.



# BI Platform Security Concepts

- Access Levels
  - There are literally THOUSANDS of rights (almost 2,500).
  - Access levels allow sets of rights to be managed as a unit.
  - There are a number of pre-defined access levels:
    - Full Control: Adds ability to create, edit, and publish documents.
    - Full Control (Owner): Full Control, but only for objects the user owns.
    - View on Demand: Adds ability to refresh a document interactively.
    - Schedule: Adds ability to schedule a document for later refresh.
    - View: View documents that have previously been refreshed.
    - No Access: Sets all rights to Unspecified.
    - Advanced:
      - Allows any combination of individual rights to be set.
      - Results in “one-off” settings that cannot be re-used.



# BI Platform Security Concepts

- Custom Access Levels
  - Can create named, reusable custom access levels.
  - Multiple access levels can be applied to the same principal / object.
  - Rights can differ based on content type ... WebI document different than Crystal Reports document, for example.
  - Individual rights can apply to objects, sub-objects, or both (i.e., can “cascade” or not).

*Custom access levels become the building blocks for a very robust, yet easily manageable security model!*



# Building Blocks Model

- Guiding principles: aka, things to (try to) avoid
  - SPARINGLY use “multiple parent” (acyclic) group structures.
    - Simplifies debugging / maintenance.
    - **Alternative is to use multiple custom access levels.**
  - AVOID breaking inheritance.
    - Simplifies debugging / maintenance.
    - **Use cascading / non-cascading rights.**
  - AVOID explicitly denying a right (leave as not specified).
    - Simplifies debugging / maintenance.
    - **Use cascading / non-cascading rights.**
  - AVOID applying granular (aka, advanced) rights.
    - Simplifies debugging / maintenance.
    - **Only use access levels.**
  - DISCOURAGE use of pre-defined access levels.
    - Blends too many concepts together, inadvertently grant more than required.
    - **Use custom access levels; consider separating application rights from content rights.**



# Building Blocks Model

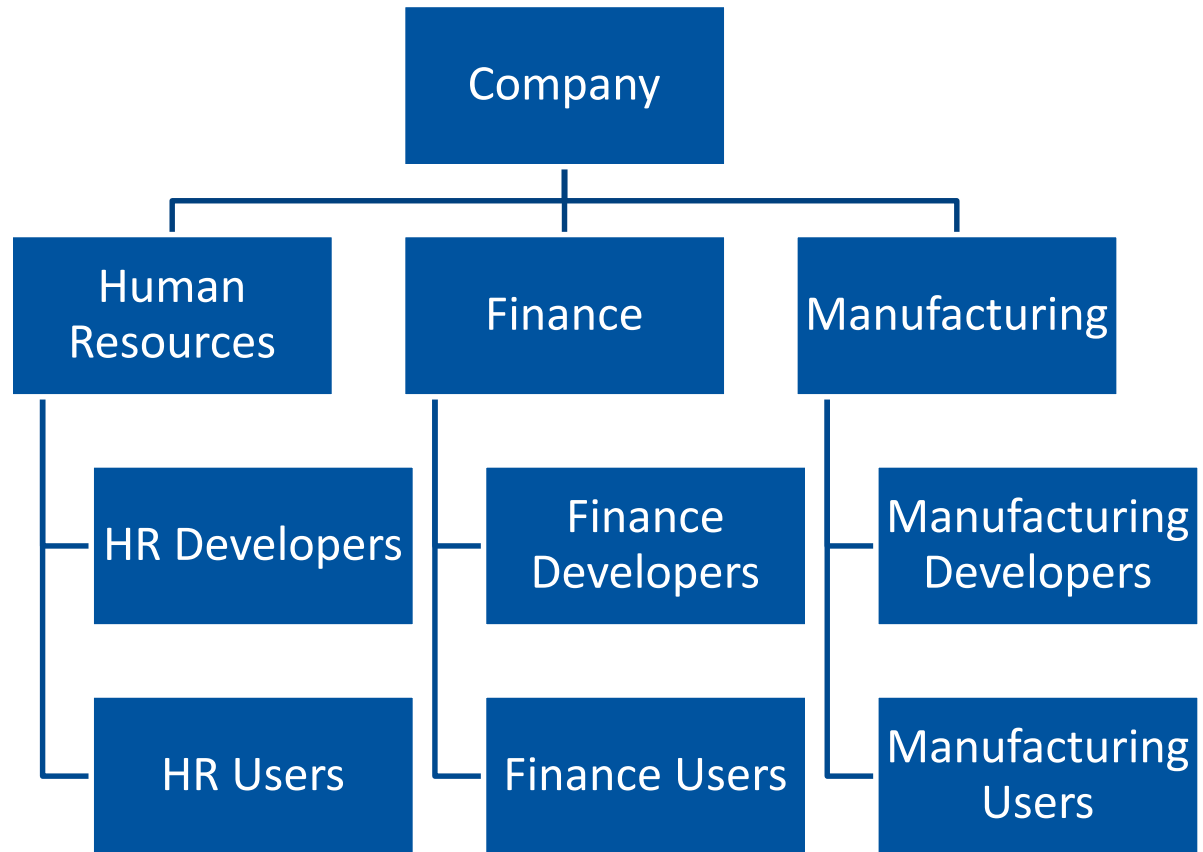
- Create building blocks: aka, things TO do
  - Take your time here! Understanding each right and a few simple principles will enable this method to work for you!
    - “MEMORIZE” the Rights appendix in the Administrator’s Guide.
  - Not all rights apply to all objects.
    - For example, schedule applies to a document object, not the Web1 application object.
    - For a given access level only include rights that relate together.
  - Some rights within the list of rights have interaction.
    - General rights can be over-ridden for specific object types (like can generally schedule, but not for Crystal Report objects).
    - Rights labeled “... that the user owns” are more restrictive than same right without that statement.
    - Rights labeled “Securely modify rights” are more restrictive than those labeled “Modify rights.”
    - For these “pairs” of rights, use one or the other, but not both.
  - THOROUGHLY review installation defaults as many are not appropriate!



# Building Blocks Model

## ■ Case Study

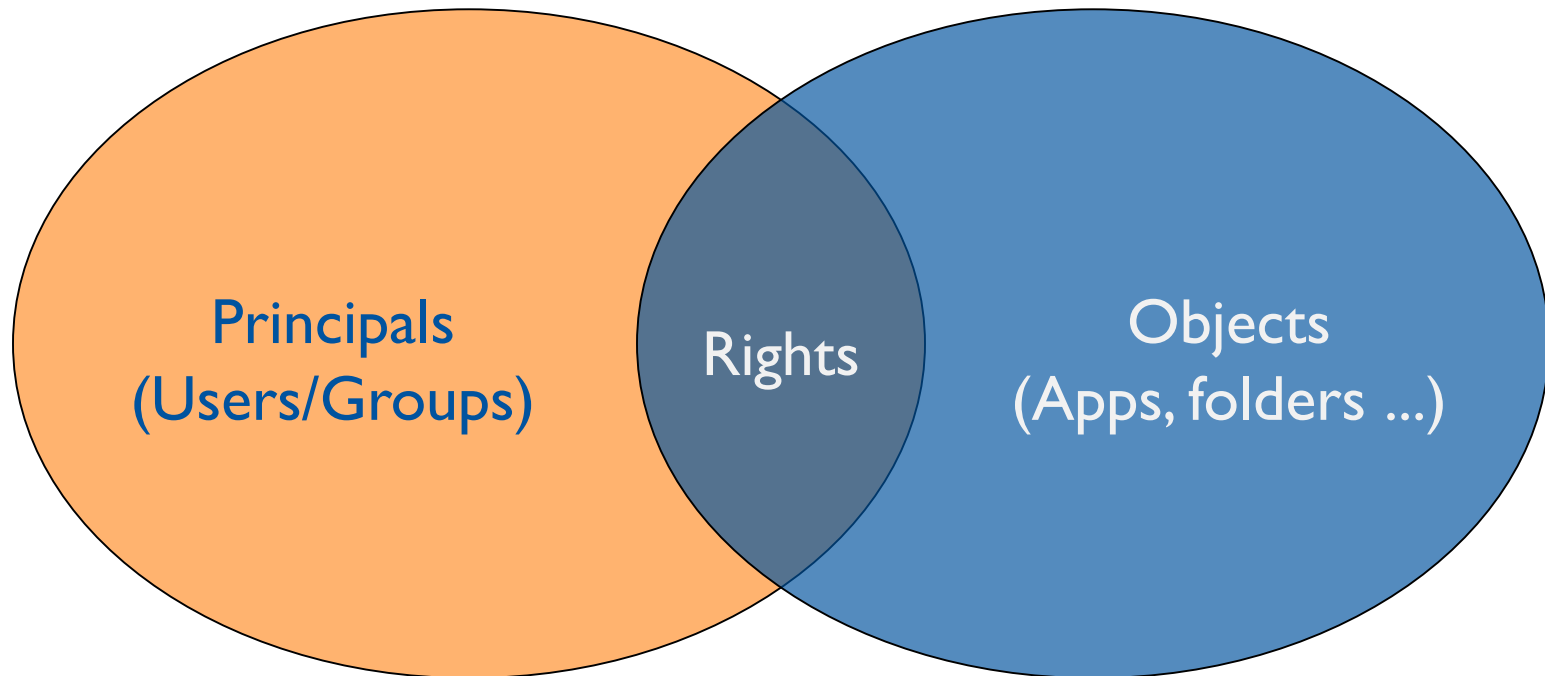
- HR, Finance, and Manufacturing each have their own data and cannot be shared.
- Web Intelligence and Information Design Tool only.
- Documents, Universes, and Connections are managed only by Developers.





# BI Platform Security Concepts

- Remember (Who, What, Where)
  - Principals (who) are given Rights (what) to Objects (where).



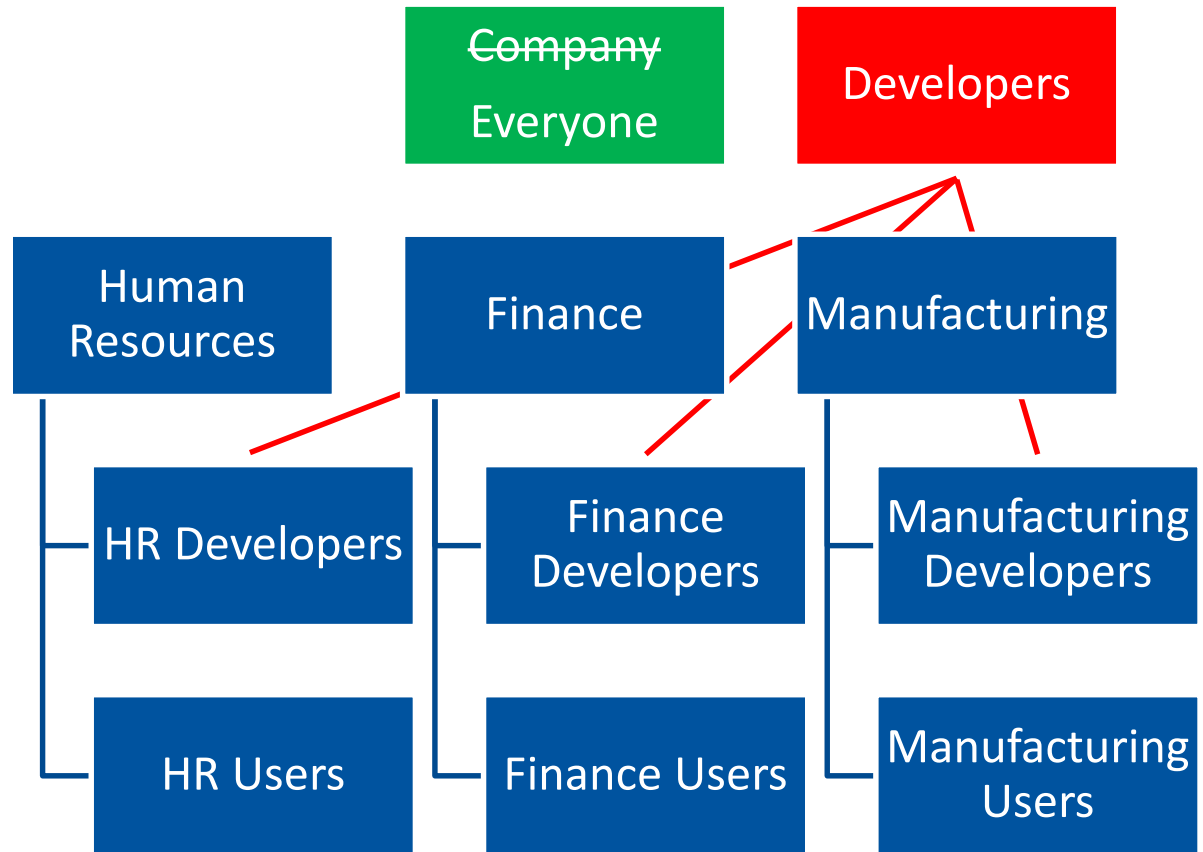
- Let's apply this to our Case Study.



# Building Blocks Model

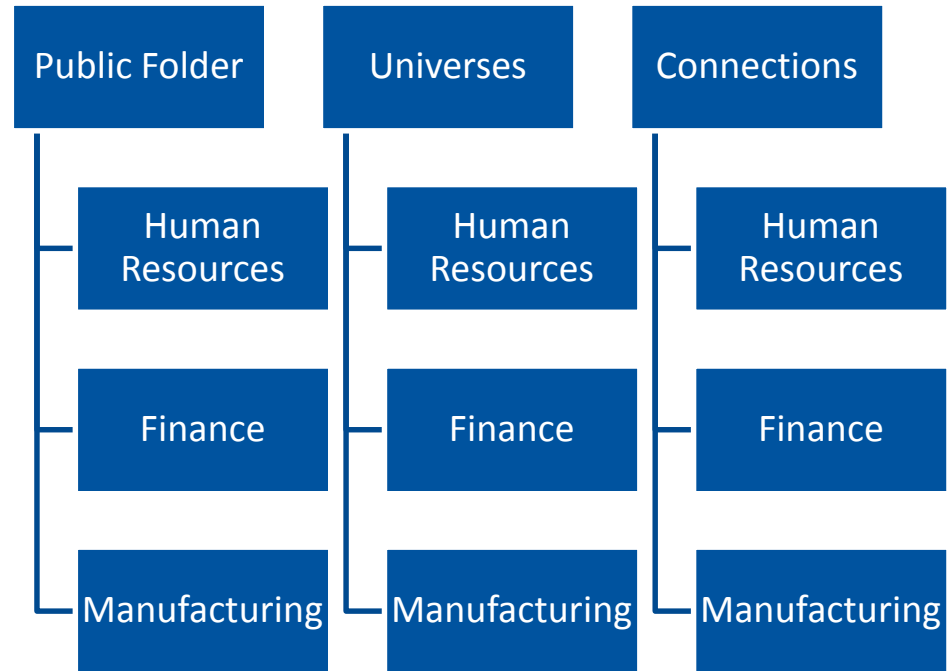
- Principals (Groups)

- Total Company will use the built-in Everyone group.
- New top-level parent group for Developers.
- Department Developer groups have two parents.



# Building Blocks Model

- Objects (Folders)
  - Public, Universe, and Connection folders for each Department.



# Building Blocks Model

- Rights (Access Levels)
  - Remember I stressed the importance of reviewing the installation defaults.
  - You will find some “surprises” that you will want to modify.
  - Let’s create a basic structure and see what happens.
  - Then we will build our first custom access level.
    - View (but not delete) of a folder, but not to any sub-objects.
    - You will want to build this one right away in your own deployment!



# Building Blocks Model

- Create Group and Folder

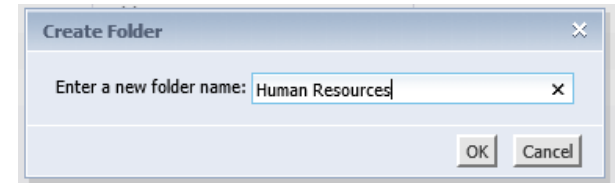
- Create Group for Human Resources.
- Create Folder for Human Resources.
- Give View rights on HR folder to HR group.
- No other rights given!



Create New User Group

Group Name: Human Resources

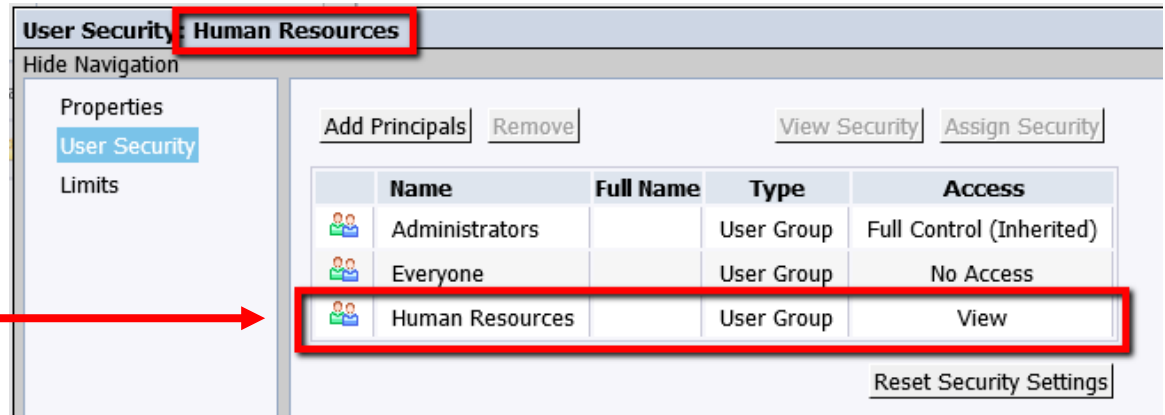
Description:



Create Folder

Enter a new folder name: Human Resources

OK Cancel



User Security: Human Resources

Hide Navigation

Properties  
User Security  
Limits

Add Principals Remove View Security Assign Security

	Name	Full Name	Type	Access
	Administrators		User Group	Full Control (Inherited)
	Everyone		User Group	No Access
	Human Resources		User Group	View

Reset Security Settings



# Building Blocks Model

- Create a New User
  - Named HR User.
  - Make a member of the Human Resources group (in addition to the default Everyone group).

### New User

Authentication Type:

Account Name:

Full Name:

Email:

Description:

#### Enterprise Password Settings

Password:   Password never expires

Confirm:   User must change password at next logon

User cannot change password

#### Connection Type

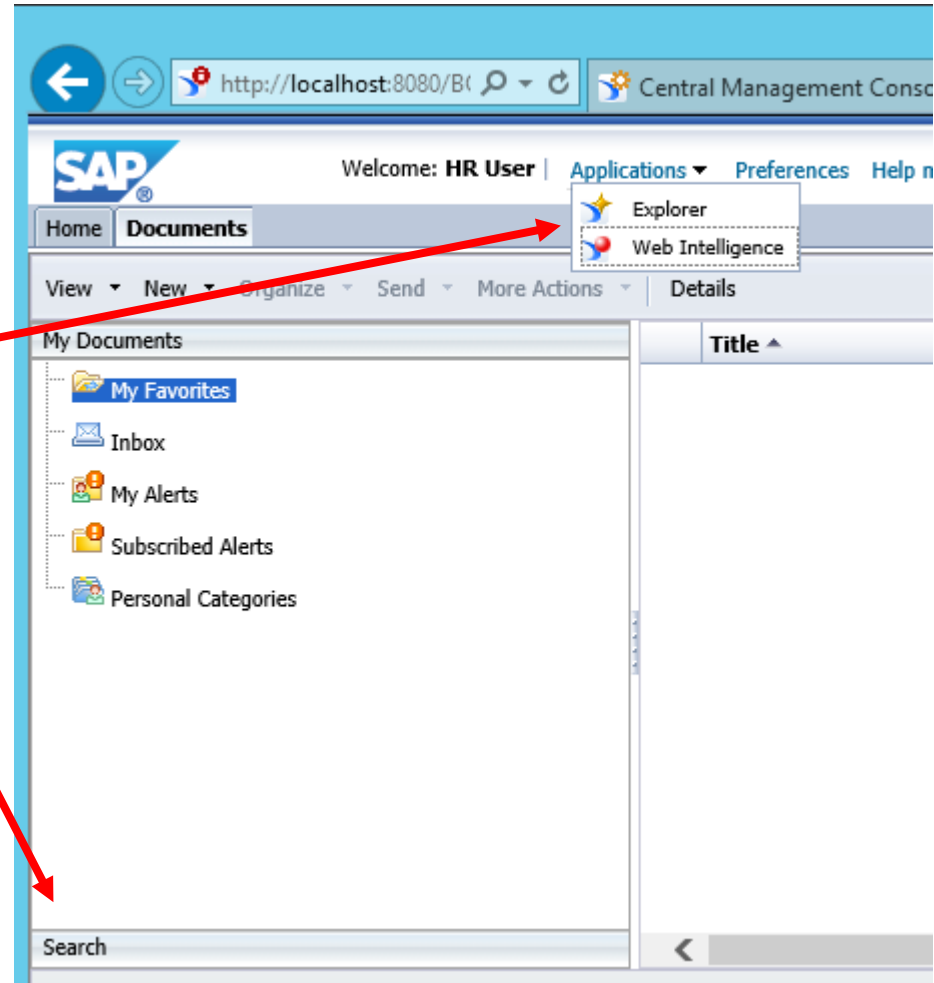
Concurrent User

Named User



# Building Blocks Model

- Log on to BI LaunchPad as HR User ...
  - Surprise!
  - Applications that weren't asked for (via Everyone).
  - No Public folders (no default top-level folder access).



# Building Blocks Model

- Provide access to root Public folder

- Create “View Folder” custom access level.
- Deny right to Delete folder (exception to the “Avoid Deny” guidance).
- Include right to View folder.
- Rights apply object only, NOT sub-objects.
- Grant “View Folder” to Everyone group for top-level Public folder.

Collection	Type	Right Name	Status	Apply To
Content	Folder	Delete objects	✘	
Content	Folder	View objects	✔	

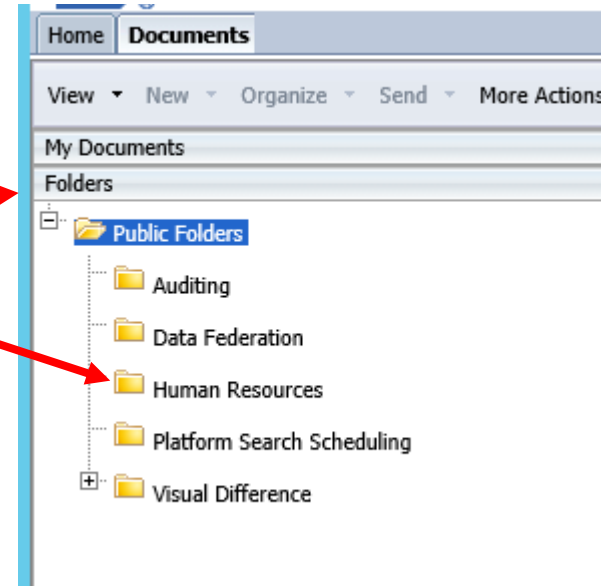
Name	Full Name	Type	Access
Administrators		User Group	Advanced
Everyone		User Group	View Folder





# Building Blocks Model

- Log on to BI LaunchPad as HR User ...
  - Can now see Public folder.
  - Can now see Human Resources folder.
  - **Surprise! Can also now see OTHER folders that weren't granted!**
  - Again, review the defaults **THOROUGHLY!**
  - There are several Everyone group defaults that probably should be removed. An Excel-based tool will be provided to assist.



# Building Blocks Model

- The built-in Access Levels may be just fine.
  - Simple to use. Can also be copied as starting points for your own custom access levels.
- These “kitchen sink” access levels may ultimately make maintenance more difficult, however.
  - The same rights may be included in multiple access levels.
  - Any changes (like adding or removing a right) have to be done multiple times.
  - If a user has more access than they should, which access level did it come from?



# Building Blocks Model

- Instead of “kitchen sink”

- Same rights in multiple places
- Then pick one access level to apply

View	Full Control
Right 1	Right 1
Right 2	Right 2
Right 3	Right 3
Right 4	Right 4
Right 5	Right 5
Right 6	Right 6
Right 7	Right 7
Right 8	Right 8

- Consider building blocks

- Each right in only one place
- Apply / inherit both where needed

Basic	Developer
Right 1	Right 1
Right 2	Right 2
Right 3	Right 3
Right 4	Right 4
Right 5	Right 5
Right 6	Right 6
Right 7	Right 7
Right 8	Right 8



# Building Blocks Model

- Create security matrix: With the “building blocks” in place, now link principals and objects to rights

Object	Principal (Group)	Access Level
Root Folders (Public, Universe, Connection)	Everyone	View Folder
Human Resource Folders (Public, Universe, Connection)	Human Resources	Basic
	HR Developers	Developer
Finance Folders (Public, Universe, Connection)	Finance	Basic
	Finance Developers	Developer
Manufacturing Folders (Public, Universe, Connection)	Manufacturing	Basic
	Manufacturing Developers	Developer
Web Intelligence	Everyone	Basic
	Developers	Developer
Information Design Tool	Developers	Developer



# Tools

- Other useful building blocks (custom access levels).
  - Create assistant administrators:
    - Password reset
    - Manage users (add / assign to groups)
    - Calendar maintenance
  - Finer control over documents:
    - Use stored procedures
    - Scheduling (separate from standard)
    - Edit SQL
  - Promotion Management:
    - Separate from Developers



- Central Management Console
  - Security query:
    - See all net rights between a principal and object.
    - See whether source is from an access level or advanced (granular) .
    - See whether right is inherited, and EXACT source of net right.
    - Results can be exported to text file.
  - Relationship query:
    - See for any given object, what other objects are related.
    - Useful for assessing impact of change to / delete of an object.



## ■ Other Resources

### ■ Internet:

- Business Objects Board (BOB <http://busobj.forumtopics.com>).
- Independent forum providing support for Business Objects products.

### ■ Excel tool (supplemental material):

- Created an Excel tool that lists all rights and the objects to which they apply.
- Includes default access levels and default security matrix.
- Can be used to “draft” custom access levels, and the basis for your own security matrix.
- Available for download ... shameless plug for BOB:  
<http://www.forumtopics.com/busobj/viewtopic.php?p=1020898>



# Recap / Q&A

## ■ Recap

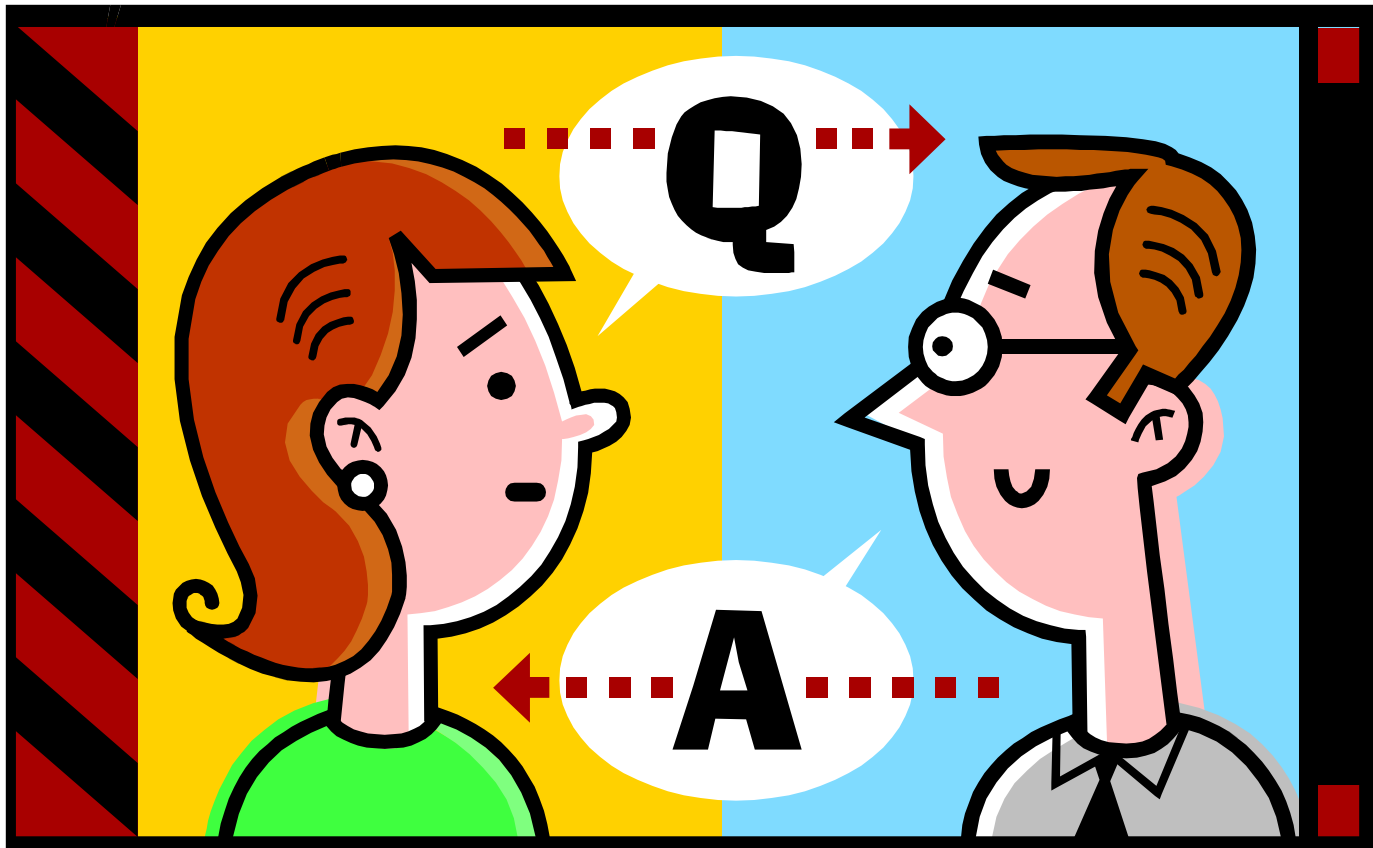
- BI Platform security concepts:
  - Principals (who) are given Rights (what) to Objects (where).
  - Inheritance / conflicting right resolution is powerful, but complex.
  - Access levels (sets of rights managed as a group) can be used as “building blocks” to minimize complexity.
- Building blocks model:
  - Review defaults, there are some you will want to remove.
  - Minimize multiple parent group structures and avoid breaking inheritance, explicitly denying rights, and granular (advanced) rights.
  - Take the time to understand rights and their interaction.
  - Create custom access levels containing only logically related rights.
  - Leverage the power of cascading / non-cascading rights.
  - Create security matrix by applying object rights to principals using custom access levels (one or more) only.
- Experiment!





# Recap / Q&A

- Q&A



# STAY INFORMED



*Follow the ASUGNews team:*

Tom Wailgum: **@twailgum**

Chris Kanaracus: **@chriskanaracus**

Craig Powers: **@Powers\_ASUG**



*Thank you for attending.*

Please provide feedback on this session by completing a short survey via the event mobile application.

**SESSION CODE  
2936**

For ongoing education in this area of focus, visit [www.ASUG.com](http://www.ASUG.com).  
Follow us **#SABOUC**



@ASUG365 #SABOUC

**ASUG**