



**rego**University

NASHVILLE • 2022

# Administration | Beginner

**Your Guides:**

Rajini Mamidi and David Zywiec

# Part I: Introduction



# Agenda

---

- Clarity Overview
- Resources and Users
- Objects and Attributes
- Security Setup
- Organizational Breakdown Structures (OBS)
- Intro to Lookups
- Financials Basics
- Scheduling Jobs

# Introductions

---

- Take 5 Minutes
- Turn to a Person Near You
- Introduce Yourself
- Business Cards

# Clarity Overview



# Clarity Overview

- Module-based system allowing customers to use features as needed
- Interconnected modules like Project and Resource allow real-time access and visibility of work being done and the resources doing the work
- Robust resource management provides the ability to find the correct resources to do the work with flexibility to accommodate any company size
- Financial module allows for costing of transactions, creating cost forecasts across a timescale, and entering non-labor costs
- Portfolio management lets management do real-time project ranking and evaluation in a pipeline and create “what-if” scenarios to evaluate future plans

# Clarity Overview #2

- Physical Components
  - Application Server
  - Jaspersoft Server
  - BG Server
  - CSA (Clarity System Administration)
  - Databases (transactional, data warehouse)
- Be familiar with the following when opening tickets with a provider
  - Clarity version
  - SQL or Oracle
  - Windows or Unix Server

# Clarity Overview #3

- The Clarity application is administered by an application administrator and can be implemented out-of-the-box but typically is configured (“customized”) to match the needs of the organization
- You must have admin rights to have access to the Administration menu
- Some common administrative responsibilities:
  - Creating/Updating Resources and Users
  - Creating/Updating Project Templates
  - Adding custom fields, configuring screen and list layouts
  - Creating and executing custom processes to fill gaps in OOTB functionality
  - Scheduling jobs
  - Monitoring application health
  - Troubleshooting and Level 2 user support

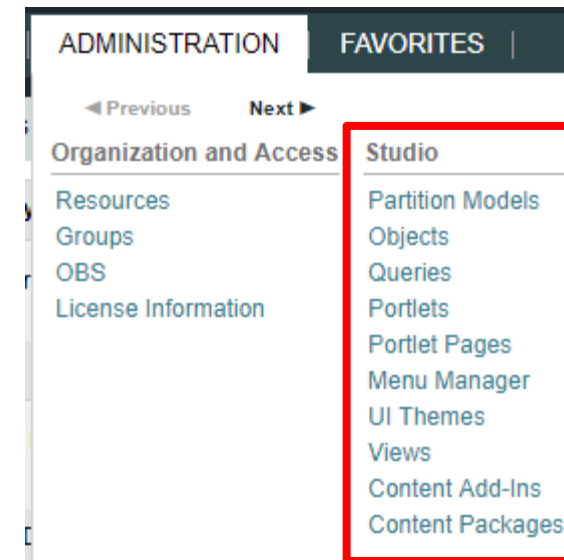


# Objects And Attributes



# Clarity Studio

- Studio is the interface used to create new or modify existing components and customize the UI to meet the needs of your organization.
- A Clarity Studio license is required to use this functionality and users must have a variety of related security rights (Administration-Studio, Create/Edit Objects, Portlets, Pages, etc).
- Studio is accessed via the Administration menu.



# Objects

---

- Objects are the major functional components of Clarity
- Objects define the attributes (fields), subpages (links), page layout, and views that make up your configured instance of Clarity
- In addition to the stock objects delivered with the system, you can create custom objects. Custom objects are essentially tables inside the database that begin with “ODF\_CA”
- Use the default objects or create custom objects and sub-objects to manage information for specific business needs
- Once you create an object, add attributes, links, and actions and set up the views

# Objects (continued)

---

- Each object has four distinct pieces you can configure
  - Properties
  - Attributes
  - Links
  - Views
- Things to remember
  - You can only delete Custom Attributes
    - Attributes need an API Attribute ID to show in the New UX
  - Adding more than 100 custom attributes to a single custom object may impact performance
  - A hierarchy with a maximum of three levels of objects can be created, and allow child objects to inherit properties and access rights from parent objects

# Objects: Types

---

- Stock Objects
  - Primary Standard Objects
    - Project
    - Task
    - Team
    - Resource
    - Company
    - Application
- Custom Objects
  - Master Objects
  - Sub-Objects

# Objects: Investment Object

---

- The Investment object is a special component that provides the ability to define attributes one time and share them across select OOTB objects
- These objects “inherit” attributes from the Investment object:
  - Project, Idea, Other Work, Application, Asset, Product, Service
- Streamlines the creation process and ensures consistency across objects
- You may re-label attributes on shared objects if needed (ID remains the same)
- Attributes defined at the investment level are available to the stock objects noted above but are not required
- You must make updates to Investment attributes **at the Investment level**

# Attributes

---

- Attributes are the fields on an object that store information
- The attributes of each object are available on the Attribute screen within the object
- Many attributes are delivered out-of-the-box, but you can create an unlimited amount of additional attributes using Studio
- Once created, you can organize and place attributes on views and portlets and use for reporting
  - Example: “Start Date” is an attribute of the project object
- Details on the various data types can be found in the Studio Developer Guide

# Attribute Data Types

---

- When creating attributes the following data types are available:
  - String (2000-character maximum)
  - Large String
  - Number
  - Calculated
  - Money (includes currency code)
  - Boolean (checkbox)
  - Date
  - Lookup (related lookup needs to be available - create prior to creating attribute)
  - Multi-Valued lookup (same note above applies)
  - Attachment
  - Time-varying
  - URL (Links to actual data)

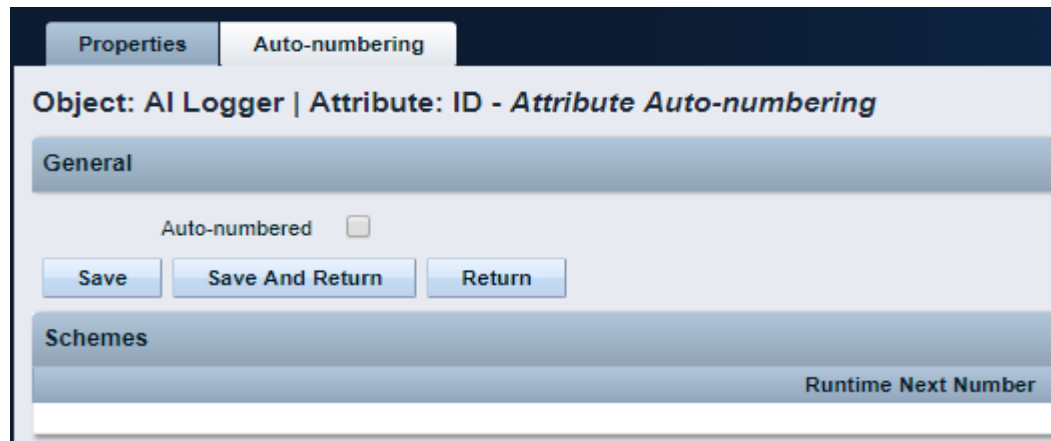


# Calculated Attributes

- A Calculated attribute displays a dynamic, read-only value
- Values are calculated from other existing attribute values
- Values are calculated every time the user accesses or refreshes the page
- These values **are not stored** in the database
- Calculated attributes can only read the following data types :
  - **Number**: Use this data type to calculate a number value like a sum or average
  - **String**: Use this data type to concatenate two or more text values
    - Example: concatenate the value of the attribute “created\_by” and the constant “2007” to produce a result of “smith 2007”
  - **Date**: Use this data type when you need to calculate dates using basic arithmetic or to provide the current date
- NOTE – You cannot delete source attributes used in a calculated attribute!

# Attributes: Auto-Numbering

- Clarity provides the ability to create your own numbering/naming scheme for object instances (PRJnnnnn, APPnnnnn, etc)
- The scheme can be numeric or a mixture of characters and numbers
- Two out-of-the-box attributes that are commonly auto-numbered are “Name” and “ID”
- Configuration is done via the Auto-numbering tab on the attribute detail



# Attributes: Exercise

- Create a new custom sub-object to the project object
  - Administration -> Objects
  - Click New and fill out the required fields

**Object Name:** Meaningful name summarizing the object function  
**Object ID:** Use a standard naming convention; many start with a customer ID + “\_” (special characters and spaces should not be used in IDs)

## Master or Subobject

- Choose **Master** if object is to be standalone
- Choose **Subobject** if object is to be accessed via another object (1 to many relationship)
  - Choose the Master object by using the browse button

Create Object Definition

Object Name

Object ID

Content Source

Description

Master or Subobject  Master  Subobject

Partition Model

Master Object

Event Enabled

Copy Enabled

Export Enabled

View All Enabled

API Enabled  (Once the value is enabled, it cannot be disabled.)

Object Extension  (An object created with an extension cannot be deleted.)

Save Save And Return Return

- Select the following checkboxes if they apply
  - **Event Enabled:** Specifies that the process engine is notified of object instances that are created or updated. (If a process needs to get driven off the object)
  - **Copy Enabled:** Specifies that copies can be made of the object instances.
  - **Export Enabled:** Specifies that object instances can be exported to XML.
  - **View All Enabled:** Specifies that the object instances can have a view containing all properties, sub-object lists, and page portlets that can be personalized on a single page.
  - **API Enabled:** Allows custom objects that are a sub object of projects/investments to show in the New UX.

# Attributes: Exercise (cont.)

- Notice the default fields included in the newly created object

Properties | Attributes | Linking | A

Object: Training Object - Attributes

Attribute Name

Attribute Display

<input type="checkbox"/>	Attribute	Descript
	Created By	
	Created Date	
	ID	
	Last Updated Date	
	Name	
	Page Layout	Page Layout
	Partition	Code that identifies the
	Updated By	

**Created By and Created Date:** Keep track of who and when the record was created.

**Last Updated By and Updated By:** Keep track of the last person who updated the record and when.

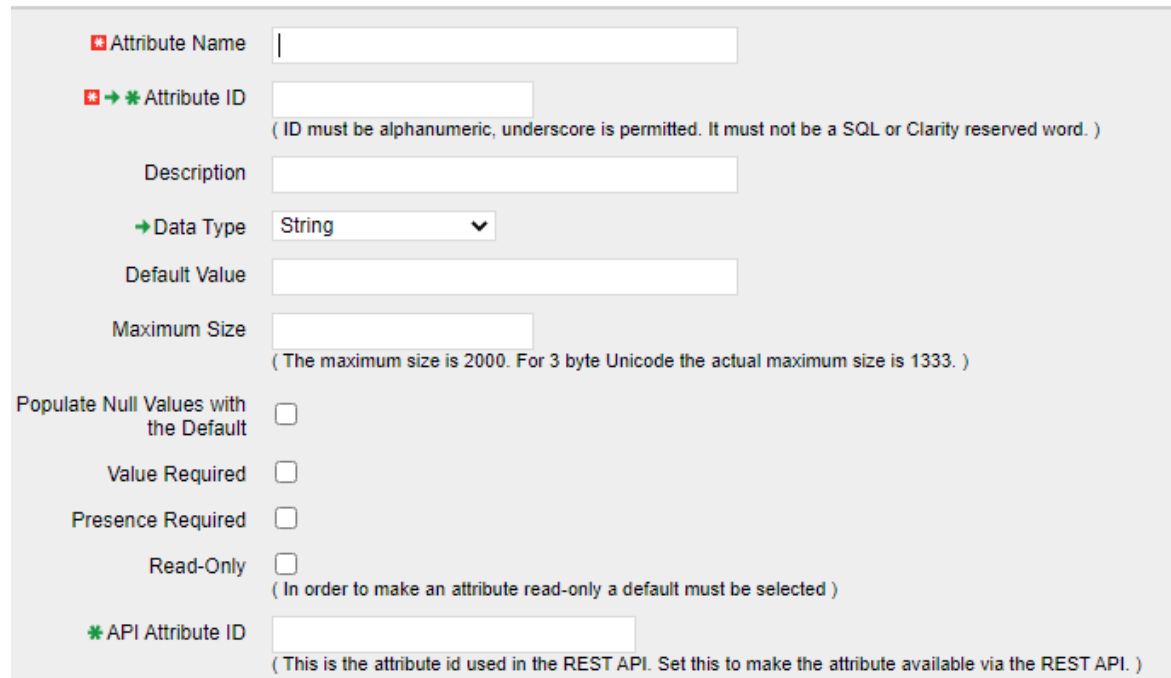
\*\* Note: Outside of custom objects there are OOTB jobs / processes that will skew the results of the last updated by and date fields, as the application often makes updates to the record.

**Page Layout:** Each object defaults to a standard layout with tabs such as Properties, Processes, and Audit. This can be customized by adding a new custom page layout. (Details later)

**Name and ID:** Used to identify the record; Name can be repeated multiple times while the ID must be unique. Auto-numbering is often used to force that uniqueness and standardization.

# Attributes: Exercise (cont.)

- Add 3 or 4 attributes of different varieties to your new custom object
  - Click New and fill out required fields as well as Data Type



The screenshot shows a form for creating a new attribute. The fields and their values are as follows:

- Attribute Name:** [Empty text input]
- Attribute ID:** [Empty text input] (ID must be alphanumeric, underscore is permitted. It must not be a SQL or Clarity reserved word.)
- Description:** [Empty text input]
- Data Type:** String (dropdown menu)
- Default Value:** [Empty text input]
- Maximum Size:** [Empty text input] (The maximum size is 2000. For 3 byte Unicode the actual maximum size is 1333.)
- Populate Null Values with the Default:**
- Value Required:**
- Presence Required:**
- Read-Only:**  (In order to make an attribute read-only a default must be selected)
- API Attribute ID:** [Empty text input] (This is the attribute id used in the REST API. Set this to make the attribute available via the REST API.)

**Attribute Name:** Name of attribute (display name can be changed later in fields if need be)

**Attribute ID:** Unique ID for attribute (spaces and special characters not allowed)

**Description:** Meaningful explanation for attribute usage

**Data Type:** Type of attribute

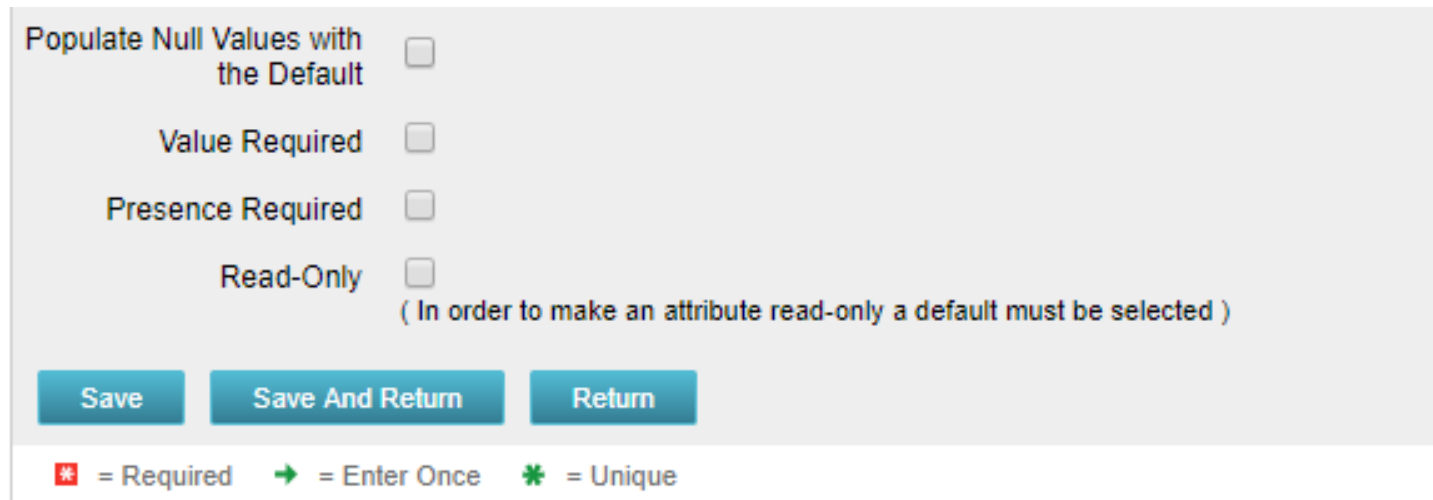
**Lookup:** Only shows up for “Lookup” and “Mutli Valued Lookup” types

**Default:** Default value to populate attribute with ( if applicable)

**API Attribute ID:** Required for field to show in New UX. Only available if Object is API Enabled

# Attributes: Exercise (cont.)

- Select the following checkboxes if they apply:
  - **Populate Null Values with the Default:** If an attribute is added after instances have already been created, this will populate the existing records with the default value
  - **Value Required:** Specifies whether a value is required for the attribute
  - **Presence Required:** Specifies that the attribute always appears in the Edit Properties view.
  - **Read-Only:** Specifies that a user cannot make changes to the value in the attribute



Populate Null Values with the Default

Value Required

Presence Required

Read-Only

( In order to make an attribute read-only a default must be selected )

\* = Required   → = Enter Once   \* = Unique

# Attributes: Exercise (cont.)

- Auto-number the ID attribute
  - Select the auto-numbering tab
  - Check the “Auto-numbered” box and click “Save”
  - Select Scheme -> Edit

Properties | Auto-numbering

Object: Training Object | Attribute: ID - *Attribute Auto-numbering*

General

Auto-numbered

Save Save And Return Return

Schemes

Partition: System ▾

Partition	Runtime Next Number	Scheme
System	00000001	[Edit]

# Attributes: Exercise (cont.)

- The default segment type is numeric, but this can be modified to include text characters as well

Object: Training Object | Attribute: ID | Partition: System - Auto-numbering Scheme

**General Information**

Scheme Name System Default

Partition System

Next Number 00000001

Status Active

**Segments**

<input type="checkbox"/>	Type of Segment	Text Value	Counter Starting Number	Counter Length	Auto-extended
<input type="checkbox"/>	Numeric Counter		00000001	8	<input checked="" type="checkbox"/>

New Delete Reorder Save Save And Return Return

= Required

- Select “New” and set Type of Segment = “Text” and type the text value into “Value” field
- Click Save and Return

**Segment Properties**

Type of Segment Text

Value TRN

Save And Return Return



# Attributes: Exercise (cont.)

Object: Training Object | Attribute: ID | Partition: System - Auto-numbering Scheme

General Information

\* Scheme Name System Default

Partition System

Next Number 00000001TRN

Status Active

Note the new numbering scheme

Segments

<input type="checkbox"/>	Type of Segment	Text Value	Counter Starting Number	Counter Length	Auto-extended
<input type="checkbox"/>	Numeric Counter		00000001	8	✓
<input type="checkbox"/>	Text	TRN		3	

New Delete Reorder Save Save And Return Return

- The new scheme defaults into the order it was entered. To reorder to have the text in front select “Reorder” and move the segments accordingly
- Select “Save and Return”

Reorder Auto-numbering Scheme Segments

Scheme Segments

Text(Next Value: TRN)

Numeric Counter(Next Value: 00000001)

Save Save And Return Return

# Resources and Users



# Resources vs Users

- Resources are managed under Home->Resources
  - Types are Labor, Equipment, Material and Expense
  - Can be allocated to projects and have costs associated with them
  - Labor resources are employees or contractors (e.g., Developer, Business Analyst)
  - Resources can be Active or Inactive
- Users are managed under Administration->Resources
  - Users are people or system IDs that log in to perform a function
  - A User record is automatically created for every Labor resource defined
  - Not all Resources need access to Clarity so those User records should be locked
  - Users can be Active, Locked or Inactive

# Resources vs Roles

- Resource – A person or thing (equipment, material, etc.) used to perform a task or capture a project expense
- Role – A generic description of a function performed by a resource
  - Examples are Developer, Business Analyst or Project Manager
  - Resources are assigned a primary (or default) role but can perform other roles
  - Used for planning and scheduling work effort
  - Roles are usually replaced by named resources when work commences
- Roles appear in the Resource List and share the same attributes

# User Status

- There are three statuses associated with a Resource on the admin side of Clarity
  - Active – The User is an active resource and can log into Clarity
  - Lock – The User is an active resource and can be assigned to projects, but cannot login
  - Inactive – The User is no longer available as a resource in the application. This is sometimes used when a resource is no longer with the company
- Additional notes on “Lock” status
  - This is a normal status for many resources (e.g., those who don’t enter time in Clarity)
  - Locked resources are still active and available in the application, just like Active resources
  - This status can be set manually by an administrator, automatically via a custom process, or automatically by Clarity if the user exceeds the valid number of login attempts (if using internal Clarity security)
  - ***Use whenever possible to keep license usage to a minimum***

# Resource Creation

- Resources can be created from either “side”
  - Application side: Home->Resources (Resource record)
  - Admin side: Administration->Resources (User record)
  - If you create a labor Resource, Clarity will generate an associated User record for you
  - If you create a User, Clarity will generate an associated Resource record for you
  - Creation is normally done from the application side since there are many more attributes to populate than on the admin side
  - When you create a Resource record the resulting new User record is set to Lock status
  - Common attributes (e.g., email address, manager) are kept in sync automatically

# Resource Creation Demo

---

- Create a Resource record
  - Application side: Home->Resources
  
- Create a User record
  - Admin side: Administration->Resources

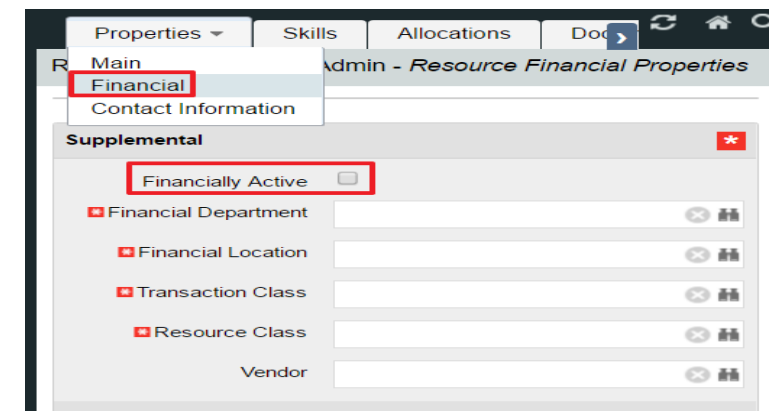
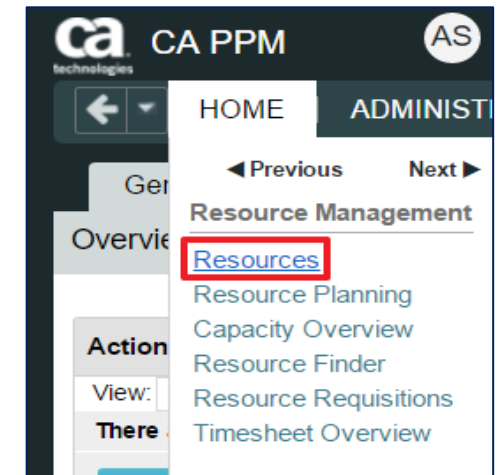
# Role Creation Demo

- Create a new Labor Role
  - Select Home > Resource Management > Resources
  - Change Is Role Filter to Yes in filter options, then Filter to display existing roles
  - Click New
  - Select Role on the Select Resource Type page
  - Select Labor for Resource Type then click Next
  - Populate the required fields then click Save
- Financially Enable a Role
  - Select role using the Is Role filter as described above
  - Select Properties > Financial
  - Check the Financially Active box and populate Transaction Class and Resource Class
  - Click Save and Return



# Class Exercise: Creating A Resource

- Create a new resource - Select Administration > Organization and Access > Resources
- Click New
- Enter user information on the Properties page
- Click Save
- Click the Properties tab, then select the Financial drop down
- Fill in the required fields
- Check the Financially Active box
- Click Save and Return



# Security Setup



# Security Considerations

- Types of Rights
  - Global – all records of a given type (all projects, all resources, etc.)
  - OBS – all records of a given type associated with a specific OBS
  - Instance – one specific record (a single project, resource, etc.)
  - Inherent – automatically assigned by Clarity
- Can be assigned at group or individual level
  - Best practice is group level
  - Define groups based on needed functionality (e.g., PM Group, RM Group, Admin group)
  - Provides option to restrict access to menu options, pages and portlets
  - Provides option to secure sub-objects and sub-pages
- NOTE – the rights you grant can impact your licensing costs!

# Security Rights

- Global Rights
  - Overrides any other associated instance or OBS level rights
  - Provides broad access to objects of a particular type
  - Allows users access a general area of the application to perform a specific function or to all instances of an object
  - A common global right is Resource – View All
  - Recommended where possible for performance reasons
- OBS-Level Rights
  - More limited access based on the OBS that a record is associated with
    - E.g., Project – View access to all projects where Department OBS = Corporate/Marketing
  - Common use case is to isolate resources or projects for discrete organizations

# Security Rights #2

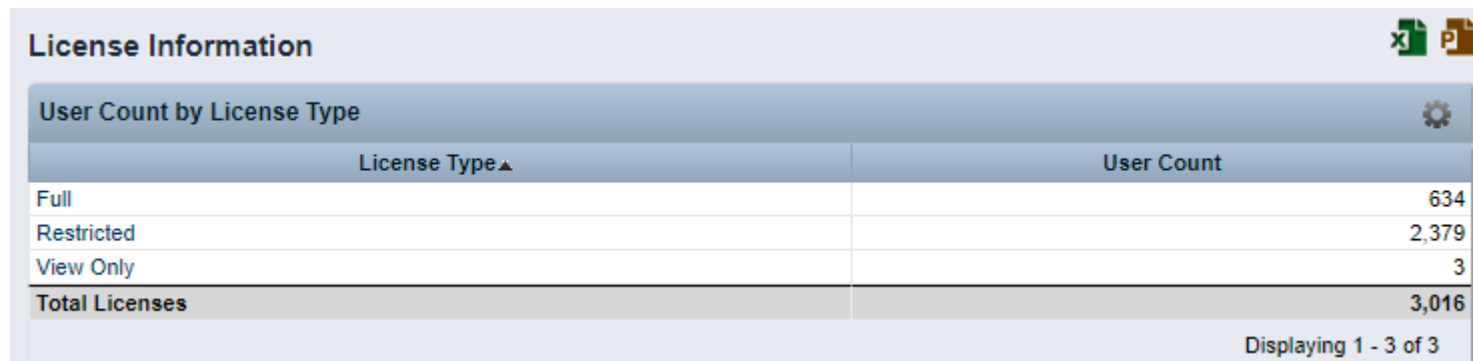
- Instance Rights
  - Provide access to a specific instance of an object (one project, one resource, etc.)
  - Can be assigned to an individual or a group
  - Can lead to performance issues if not managed carefully
  - ***Difficult to manage*** - minimal use is recommended
- Inherent/Automatic Rights
  - Automatically applied when user is first added to Clarity or the user is associated with a particular field on an object (e.g., Resource Manager)
  - Irrevocable
    - User Favorites Menu – Edit, Project – Manager (Auto)
  - Revocable
    - Resource – Enter Time (for themselves)

# Security Groups

- Groups
  - A set of users who perform similar functions or roles in the tool
  - Contains a collection or combination of rights applicable to each member of that group
  - Can associate a single resource with any number of groups
  - Can contain a collection of Instance, OBS, and Global level rights
  - Quickest way to assign multiple rights to several people at once
  - Much easier to manage than assigning rights to specific individuals
- Other Uses for Groups
  - Can be used to determine what users see in the application
    - Menu options
    - Pages, Portlets and Tabs
  - Groups can also be created for things other than granting rights
    - E.g., notifications from a workflow

# Security and Licenses

- The rights assigned to an individual or group determine the type of license consumed and counted by Clarity
- Use the License Information portlet (Administration->License Information) to monitor usage and maintain compliance
- Audit this usage periodically



License Type ▲	User Count
Full	634
Restricted	2,379
View Only	3
<b>Total Licenses</b>	<b>3,016</b>

Displaying 1 - 3 of 3

# Security Best Practices

- Assign people their rights only once
  - Granting the same rights at different levels or through multiple groups can slow down performance in some pages/views
- Only use Instance rights as a last resort and try to keep at a Group level
- Minimize security maintenance
  - It's better to create more Groups with a relatively small number of rights than to create a few Groups with rights to many different objects.
  - This approach makes it easier to add and remove rights when people change roles, you can simply add them or remove them from the appropriate groups
- Understand what each right means
  - Know the definition of each right
  - Understand the licensing implications of different rights



# Security Best Practices #2

- Think about security “roles”
  - Not the same as Primary Role
  - Theoretical, not in the system (examples: Timesheet user, Project approver)
- Document your security model design before building it
  - Use OOTB groups as a reference but most likely you will create your own groups
- Set up the model in Clarity
  - Create dummy data
  - Create generic test users – one per functional “role”
- Log in as each test user and verify the model is working as designed
  - Can you see and do what you expect for each role?
  - Can you see or do anything that shouldn't be allowed?

# Security Exercise

- Select Administration->Groups
- Click New
- Complete Required fields (\*ID must be unique)
  - For class, use your name (e.g., Angie's Group)
- Click Save and Continue
- Hover over the Groups Access Rights Tab
- Choose Global, click Add
- Use the Filter section to search for and add the following:
  - Timesheets – Navigate
  - Resource – Navigate
- Click on the Resources Tab, Click Add
- Search for a resource and click Add

# Organizational Breakdown Structures (OBS)



# Organizational Breakdown Structure (OBS)

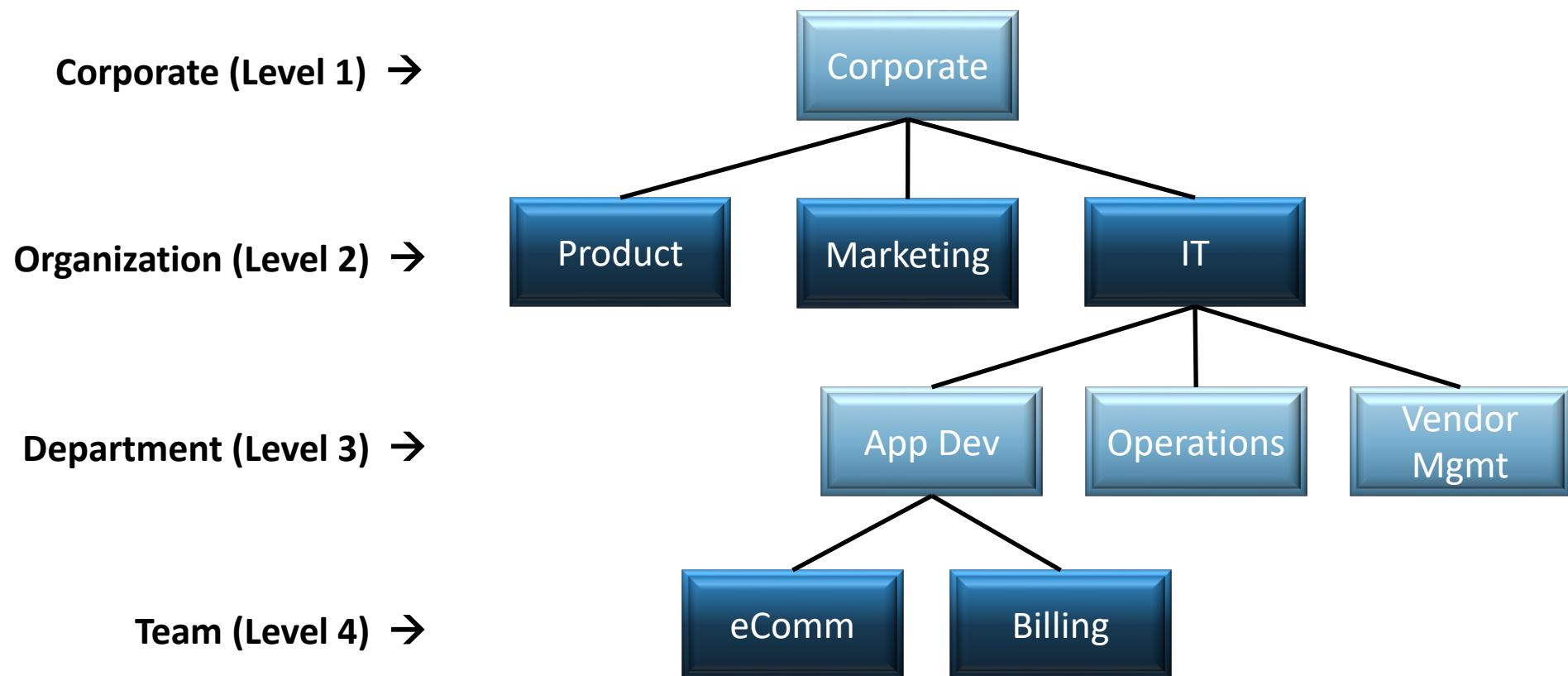
- An OBS is a company-defined hierarchical structure to categorize projects and resources
- Can be used for filtering, reporting and security (OBS rights)
- You can create any number of OBSs with up to 10 levels
- Security can be enabled for any OBS
  - Department OBS and Location OBS only on initial creation (until tied to Entity)
- Remember that every OBS created will need to be maintained which can be a time-consuming task depending on complexity
- OBS is a default filter parameter on most out-of-the-box reports

# OBS Types

- Identifying what your business needs are is the first step in determining what types of OBS structures to build
  - In what ways do we need to group data for portlets and reports?
  - Are there other fields that can be used to accomplish this? (e.g., Sponsoring BU)
  - Is there a need for multi-level filtering or roll-up reporting?
- Some OBS Examples:
  - Organizational (like HR org chart)
  - Financial Departments / Cost Centers
  - Functional Organizations and Teams
  - Product or Product Teams

# OBS Example

- Functional/Team OBS



# OBS Maintenance

- Maintaining an OBS and associated records can be time-consuming
- Keep the number and complexity of OBSs to a minimum
- Changing the structure and/or labels in an OBS can affect reporting
- Deleting an OBS unit will also delete all child units under that level
- Any record tied to a deleted OBS unit will be “orphaned” (null value in OBS field)
- Units can be moved from one parent to another within a specific OBS and all child units will follow
- OBS associations can be done using a list view, but the fastest method could be from the Admin menu (OBS Unit tab “Attached Instances”)
- Look for automation opportunities (e.g., do associations via GEL script)

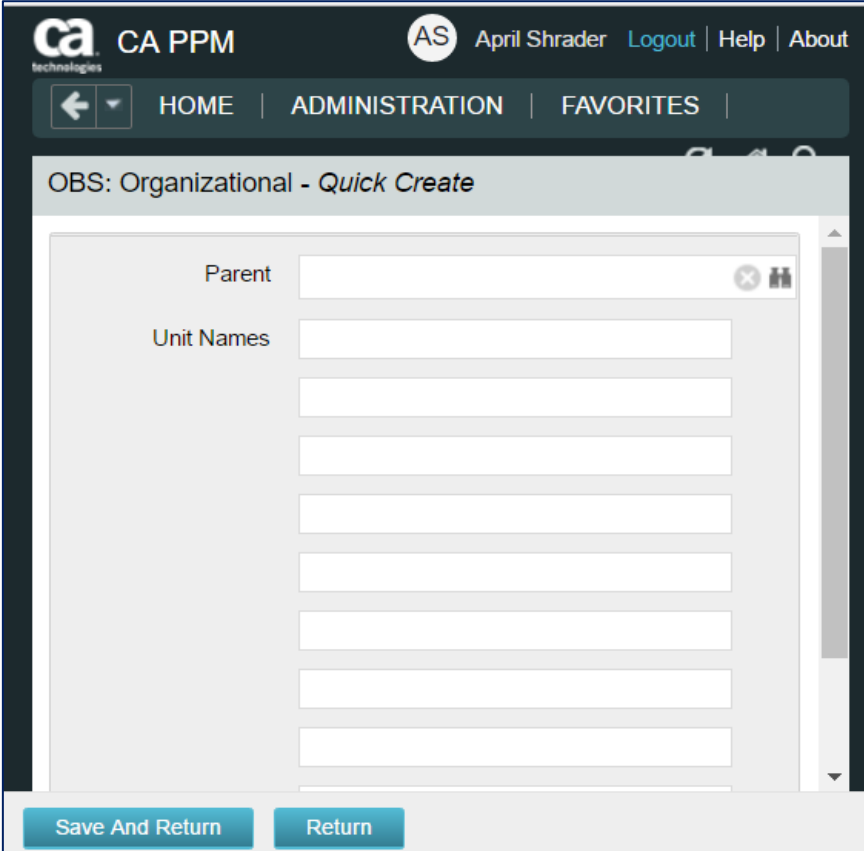
# OBS Demo #1

- Create an OBS
  - Select Administration->OBS, click New
  - Name the OBS (this is what users see)
  - Create a unique OBS id (for example, xx\_ex\_obs)
  - Name the levels of the OBS (for example Department, Location)
  - Choose the Associated Objects this OBS will be associated with (e.g., Project object)
  
- Add Units to the OBS
  - Click on the Units tab
  - Click New
  - Name the Unit and choose the Parent Level
  - Use the Quick Create feature to speed up the process



# OBS Demo #2

- Use Quick Create to add multiple units under the same parent
  - Click the Units tab and then Quick Create
- Attached Instances – assign records to the OBS
  - Click on the **Attached Instances tab**
  - Choose the **Object** from the dropdown
  - Any records already attached will display
  - Click **Add** to add additional records to the Unit



The screenshot shows the CA PPM interface for the 'OBS: Organizational - Quick Create' form. The top navigation bar includes the CA PPM logo, the user name 'AS April Shrader', and links for 'Logout', 'Help', and 'About'. Below the navigation bar are tabs for 'HOME', 'ADMINISTRATION', and 'FAVORITES'. The main form area has a title 'OBS: Organizational - Quick Create' and contains the following fields:

- Parent:** A text input field with a close button (X) and a help icon.
- Unit Names:** A vertical stack of eight text input fields for entering unit names.

At the bottom of the form, there are two buttons: 'Save And Return' and 'Return'.

# OBS Demo #3

- Add a Resource to an OBS (3 ways to do this)
  - On the Resource record (Home->Resources)
  - On the User record (Administration->Resources)
  - On the OBS definition (Administration->OBS) – Attached Instances tab
  
- Add OBS-Level Security
  - Administration->Groups
  - Groups Access Rights->OBS Unit
  - Click Add and find the desired security right
  - Click Add and Continue and select the OBS unit
  - Click Add

# Intro to Lookups



# Lookups

- A Lookup is a list of pre-defined values that users can select from
- Lookups are attached to fields in an object's configuration (Admin->Studio->Objects)
- Many system-defined lookups are included, but some are not editable
- Clarity admins can define new lookups as needed
- Pros:
  - Improved data consistency as users are limited to a set of choices
  - Enhanced capabilities for filtering and reporting
  - Helps users understand the meaning or purpose for a specific field
- Cons:
  - Requires more planning and management to maintain an effective and complete list
  - Less flexibility in the data that can be entered in a field

# Lookup Types – Static List

- Static List
  - Simple list of values
  - The order of the values can be maintained manually by an admin, or set automatically by the system in alphanumeric order
  - Values can be added but not deleted
  - Deactivate a value to prevent it from appearing in the UI
  - The list can be a dropdown (pull-down) or a browse window - use the [Fields] link on the object configuration to set it:

View	Category	Setup	Modified
Project Properties	Properties	[Layout: Create] [Layout: Edit] [Actions Menu] [Fields]	✓
Program Properties	Properties	[Layout: Create] [Layout: Edit] [Actions Menu] [Fields]	✓
Project List	List Column	[Layout] [Options] [Aggregation] [Actions Menu] [Fields]	✓

Object: Project | Partition: System | View: Project Properties - *Property Field*

Attribute percent\_calc\_mode

Data Type Lookup - Number

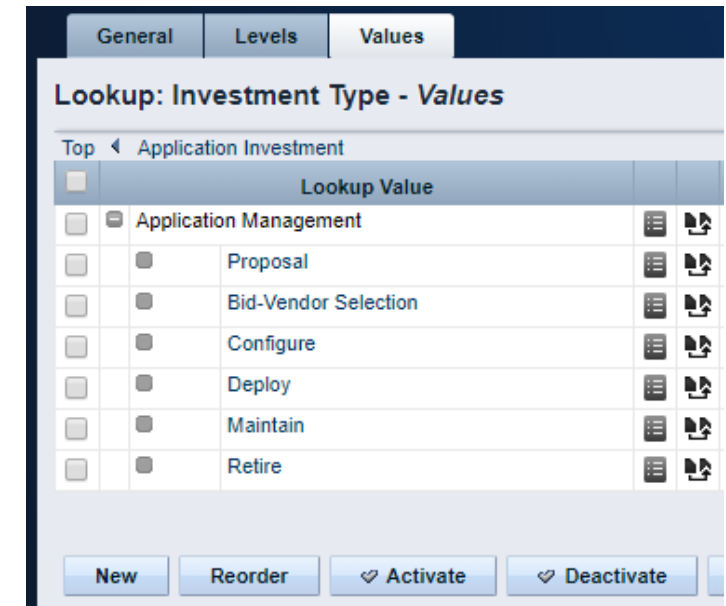
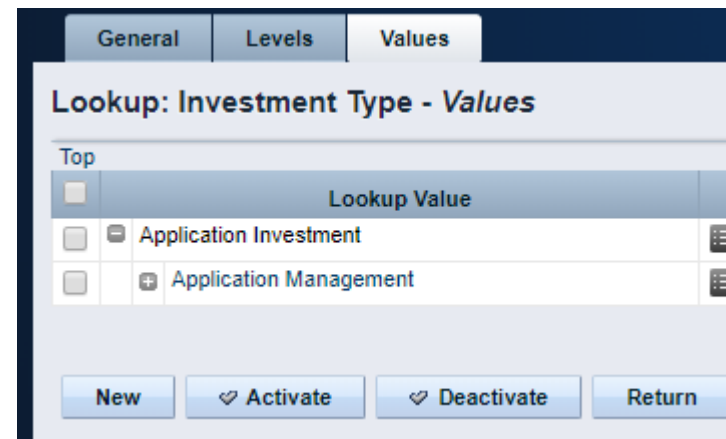
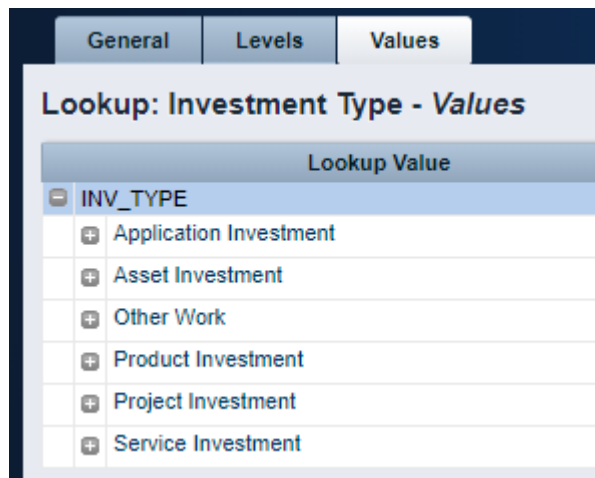
Property Label % Complete Calculation

Display Type Pull-Down

Hint Changing this method will affect how Task

# Lookup Types – Static Dependent List

- Static Dependent List
  - A multi-level list of values
  - Similar options as Static List except it can only be displayed in a browse window
  - Useful for guiding users through a long list of values that can be logically grouped



# Lookup Types – Dynamic Query

- Dynamic Query
  - Selection list is built from a real-time query of the Clarity database
  - Ideal for large lists or for lists that change frequently, some examples:
    - All Active Projects
    - All Active Managers
    - All Departments
    - Fiscal periods for current year
  - Requires some knowledge of Clarity database schema and SQL language
  - Queries are written in NSQL, an extension of SQL specific to Clarity
  - Out-of-the-box dynamic queries are a good source of information but should not be modified, create a custom copy instead.

# Financials Basics





# Financials – Considerations

- Keep as simple as possible
- Invest time to develop the right architecture
  - Include PMO, finance, PM's
  - Start with desired outputs and work backwards to build architecture
- Clarity is not the financial system of record
- Clarity is not an accounting tool
- Provide enough financial information to make decisions

# Financials – Basic Requirements

- Entity
  - Boundary for a unique set of departments, locations, fiscal periods, etc.
- Fiscal Time Periods
  - Units for reporting and financial processing (e.g., Weekly, Monthly, Quarterly, Annually)
  - Not the same as Time Reporting Periods
- Departments / Department OBS
  - Represent units in the organizational structure of the company
- Locations
  - Represent geographical locations where a company conducts its business
  - Typically linked to departments for resource and investment assignment
- Rate Matrix
  - Used during financial processing to determine cost and billing rates
  - Assigned columns identify the criteria used to match rates and costs to transactions
- Scheduling Financial Jobs

# Financials – Fiscal Time Periods

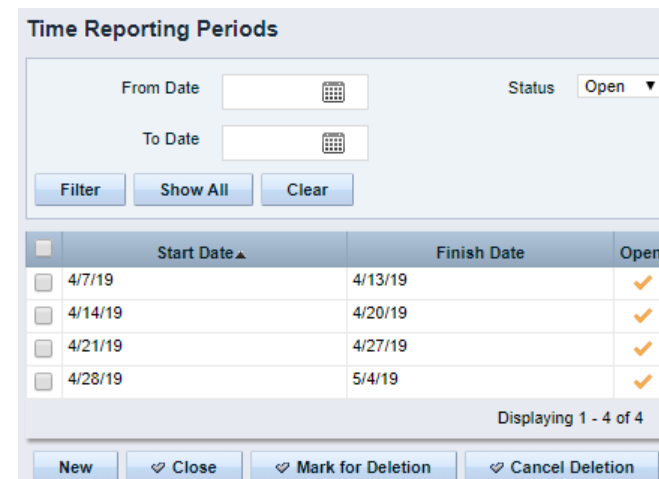
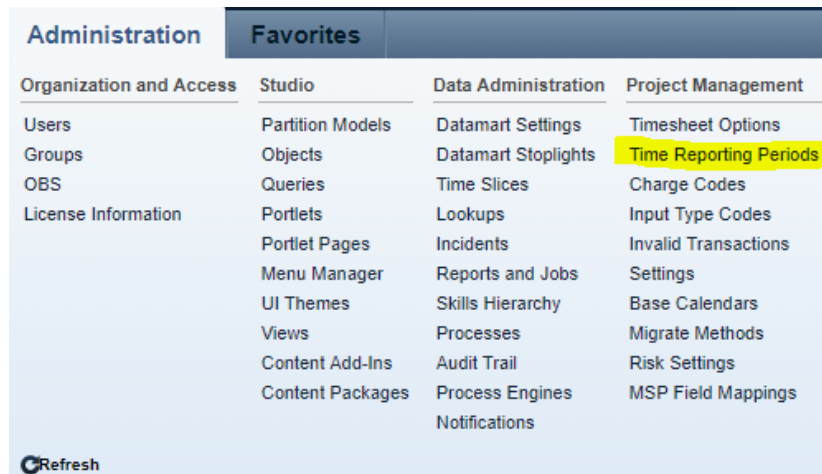
- Created by administrators, often an annual activity
- Required for financial plans
- Used in portlets and reports for financial data
- Not the same as time reporting (timesheet) periods
- Defined at the entity level under Administration->Finance->Setup->Entities

Administration	Favorites				
Organization and Access	Studio	Data Administration	Project Management	General Settings	Finance
Users	Partition Models	Datamart Settings	Timesheet Options	System Options	Processing
Groups	Objects	Datamart Stoplights	Time Reporting Periods	Site Links	WIP Settings
OBS	Queries	Time Slices	Charge Codes	Client Downloads	<a href="#">Setup</a>
License Information	Portlets	Lookups	Input Type Codes	Feature Enablement	Cost Plus Codes
	Portlet Pages	Incidents	Invalid Transactions	Data Warehouse OData Service	Manage Matrix
	Menu Manager	Reports and Jobs	Settings	Integrations	GL Accounts
	UI Themes	Skills Hierarchy	Base Calendars		
	Views	Processes	Migrate Methods		
	Content Add-Ins	Audit Trail	Risk Settings		
	Content Packages	Process Engines	MSP Field Mappings		
		Notifications			

Financial Organizational Structure	
Organizational Structure	
	Defaults
	Entities
	Locations
Transactions	
	Vendors
Currency	
	Currency
	Foreign Exchange Rates

# Financials – Time Reporting Periods

- Weekly periods for timesheets, normally starting on Sat, Sun or Mon
- Administrators create and open/close time periods per accounting needs
- Multiple weeks can be created at a time (NOTE: they are **open** by default)
- Use caution - periods cannot be changed once a timesheet is posted!



# Financials – Classification Options

Financial reporting needs should drive which classifications you will use

- Resource Classes – type of resource (Labor, Equipment, Onshore, Offshore, Executive, Staff, etc.)
- Company Classes – clients or lines of business within your organization
- Transaction Classes – user-defined values that group transaction types
  - Examples are Labor, Hardware, Software, Consulting
- Investment Classes – categorize work logically within an organization (rarely used)
- Input Type Codes – represents a breakdown of work associated with resources
  - Can be used to determine the rates and costs applied to financial transactions
  - Examples are billable/non-billable, exempt/non-exempt, regular time/overtime
- Charge Codes – represents a breakdown of work associated with investments
  - Can be used to determine the rates and costs applied to financial transactions
  - Examples are capital/expense, billable/non-billable, project phase
- Cost Type – used in Financial Plans for grouping (Capital or Operating)

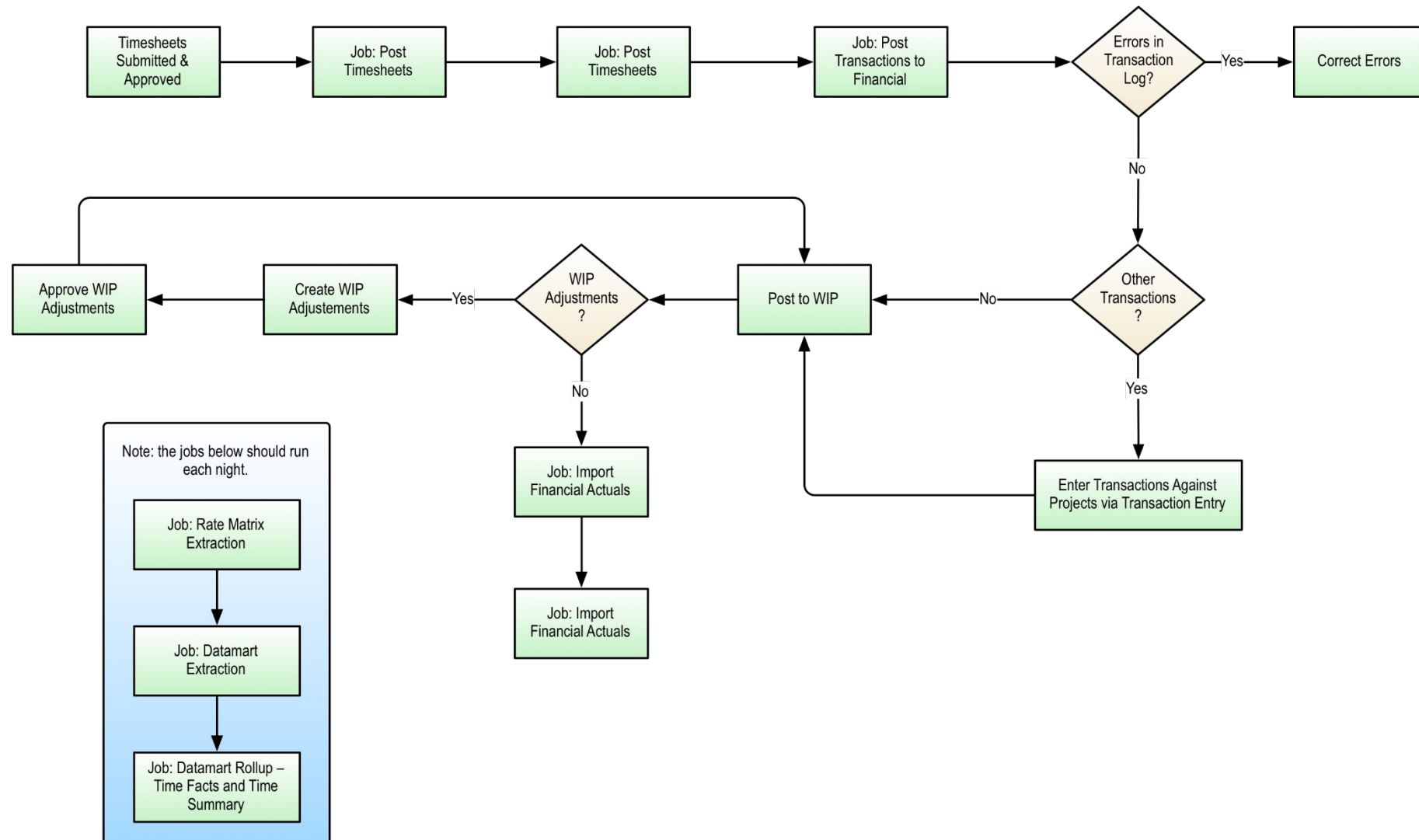
# Financial Jobs

- Clarity provides several jobs for processing financials
- Common daily schedule:
  1. Post Timesheets
  2. Post Transactions to Financial
    - Job may flag invalid transactions that should be reviewed and fixed as needed
  3. Post to WIP
  4. Import Financial Actuals
- Jobs that should only be run off-hours
  - Rate Matrix Extraction
  - Datamart Extraction
  - Datamart Rollup – Time Facts and Time Summary

Administration	Favorites		
Organization and Access	Studio	Data Administration	Project Management
Users	Partition Models	Datamart Settings	Timesheet Options
Groups	Objects	Datamart Stoplights	Time Reporting Periods
OBS	Queries	Time Slices	Charge Codes
License Information	Portlets	Lookups	Input Type Codes
	Portlet Pages	Incidents	Invalid Transactions
	Menu Manager	Reports and Jobs	Settings
	UI Themes	Skills Hierarchy	Base Calendars
	Views	Processes	Migrate Methods
	Content Add-Ins	Audit Trail	Risk Settings
	Content Packages	Process Engines	MSP Field Mappings
		Notifications	

Refresh

# Financial Process Flow



# Scheduling Jobs





# Scheduling Jobs - Overview

- Clarity provides many built-in jobs that are run in the background to perform key data-management tasks and to improve system performance
- Some examples are:
  - Financial jobs (e.g., Post Transactions, Post to WIP)
  - Data Warehouse jobs
  - Rate Matrix Extraction
  - Time Slicing
  - Synchronize portfolio investments
  - Delete/Purge jobs (e.g., audit trail, job history, process instances)
  - **Execute a Process** – commonly used by admins to run custom scripts (processes)

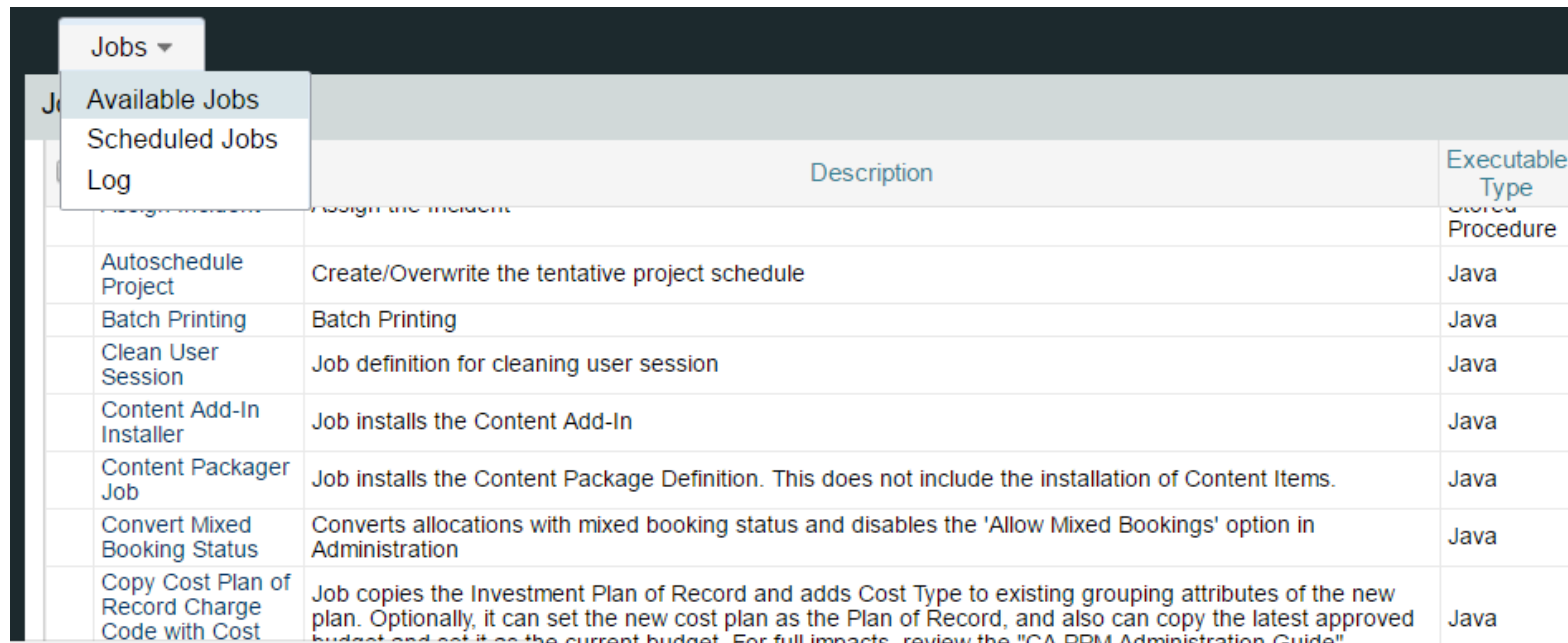
# Scheduling Jobs - Access

- Jobs are accessed under the Home menu (Reports and Jobs)
- Access is often restricted to administrators but can be granted to others using the “Jobs - xxx” security rights



# Available Jobs

- The “Jobs” tab allows you to see what jobs are available in the system and the jobs that are currently scheduled to run
- The Log option in the dropdown provides some historical job details but you will usually only see jobs that were run by you (even admins can’t see all log info)



Job Name	Description	Executable Type
Assign Incident	Assign the incident	Stored Procedure
Autoschedule Project	Create/Overwrite the tentative project schedule	Java
Batch Printing	Batch Printing	Java
Clean User Session	Job definition for cleaning user session	Java
Content Add-In Installer	Job installs the Content Add-In	Java
Content Packager Job	Job installs the Content Package Definition. This does not include the installation of Content Items.	Java
Convert Mixed Booking Status	Converts allocations with mixed booking status and disables the 'Allow Mixed Bookings' option in Administration	Java
Copy Cost Plan of Record Charge Code with Cost	Job copies the Investment Plan of Record and adds Cost Type to existing grouping attributes of the new plan. Optionally, it can set the new cost plan as the Plan of Record, and also can copy the latest approved budget and set it as the current budget. For full impacts, review the "CA PPM Administration Guide"	Java

# Running a Job

- Jobs can be run immediately, run at a future time, or run on a recurring schedule
- Click on an Available Job to see the execution options
  - After submitting a job, Clarity will display the “Scheduled Jobs” screen below

**Job Type: Delete Investments - Job Properties**

**General**

Job Name

**When**

When  Immediately  Scheduled

Start Date

Start Time

Recurrence

[Set Recurrence]

**Jobs: Scheduled Jobs**

Job Name

Job Type

Job ID

Category

Job Status

From Scheduled Date

To Scheduled Date

Recurrence

<input type="checkbox"/>	Job▲	Job Type	Job ID	Job Status	Scheduled
<input type="checkbox"/>	Delete Investments	Delete Investments	7160667	Scheduled	3/31/19 6:30 PM

Displaying 1 - 1 of 1

# Recurring Jobs

- Recurring jobs are configured using the “Set Recurrence” link at the bottom of the Job Properties page


**Job Type: Delete Investments - Job Properties**

**General**

★ Job Name

**When**

★ When  Immediately  
 Scheduled

★ Start Date  

★ Start Time

Recurrence

[\[Set Recurrence\]](#)

# Scheduling Options

- Three options are available for setting a schedule
  - Weekly – specify weekdays and months
  - Monthly – enter specific days of the month (and months)
  - UNIX Crontab
    - Provides more flexibility and granularity
    - Allows you to specify down to the minute and multiple runs per hour
    - Seems cryptic until you get the hang of it

The screenshot displays a scheduling configuration interface with the following elements:

- Recurrence** (selected):  Recurrence
- Run Once** (selected):  Run Once
- Weekly** (unselected):  Weekly
- On** (checkboxes):  Sunday,  Monday,  Tuesday,  Wednesday,  Thursday,  Friday,  Saturday
- Months** (dropdown): January, February, March, April, May, June
- Recur Until** (calendar icon)
- Monthly** (unselected):  Monthly
- Days of the Month (1-31)** (text input)
- Months** (dropdown): January, February, March, April, May, June
- Recur Until** (calendar icon)
- Use UNIX crontab entry format** (unselected):  Use UNIX crontab entry format
- Legend**:  = Required

# Scheduling With Crontab

- UNIX Crontab format requires 5 values, separated by blanks
- An asterisk (\*) means “all” (every month, every day, etc.)

Minute	Hour	Day of the Month	Month	Day of the Week
0-59	0-23	1-31	1-12	0-6 (0=Sunday)

- A forward slash denotes an interval (e.g., \*/15 - every 15 mins)
  - Examples
    - 45 6,18 \* \* \* (every day at 6:45am and 6:45pm)
    - 0 \*/2 \* \* \* (every 2 hours on the hour)
    - 30 4 \* \* 0 (every Sunday at 4:30am)
    - 30 20 \* \* 1-5 (every Mon-Fri at 8:05pm)

# Tips for Scheduling Jobs

- Use Save Parameters
- Save Jobs as a Favorite
- To avoid re-defining a job, schedule it in the future and pause it
- Use the Notify options to learn of failures
- Remove your name from Notify on Completion if you get tired of emails
- Execute a Process always runs successfully (you must check the process)



# Class Exercise – Schedule a Job

- Select Home->Reports and Jobs
- Click Jobs tab and select Available Jobs
- Scroll to the Post Timesheets job and click on it
- In the When section uncheck Immediately and check Scheduled
- Select a Start Date and Start Time ***in the future***
- Click [Set Recurrence]
- Set your desired schedule
- Click Save And Return
- Click Submit and find the job on the Scheduled Jobs screen

# Questions?



# Questions?



# Thank You For Attending regoUniversity

## Instructions for PMI credits

- Access your account at [pmi.org](https://pmi.org)
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Provider = **Rego Consulting**
- Class Name = **regoUniversity**
- Course **Description**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**



Let us know how we can improve!  
Don't forget to fill out the class survey.



### Phone

888.813.0444



### Email

[info@regoconsulting.com](mailto:info@regoconsulting.com)



### Website

[www.regouniversity.com](http://www.regouniversity.com)