OnBase Guide - WorkView Configuration

**Goal:** To configure successful WorkView Applications within the CU OnBase environment

**Complexity Level:** Departmental WorkView Administrators

2/24/2022
# Table of Contents

- **Background** ........................................................................................................... 3
- **Prerequisites** ........................................................................................................... 3
- **Required Trainings** ................................................................................................ 3
- **Resources** ................................................................................................................ 3
- **User Groups & Access** ............................................................................................ 4
  - Functional & WorkView User Groups ........................................................................ 4
  - MANAGER Group ....................................................................................................... 4
  - Master Security .......................................................................................................... 4
- **WorkView Configuration Guidelines** ...................................................................... 5
  - Classes ....................................................................................................................... 5
  - Filters and Filter Bars ................................................................................................. 8
  - Views .......................................................................................................................... 9
- **Application Creation Excelerator** ........................................................................... 9
- **Testing** ...................................................................................................................... 9
- **Importing Data to Classes** ..................................................................................... 10
- **Migration** ................................................................................................................ 10
- **Purging Objects** ...................................................................................................... 10
  - Purge Objects Selectively (Recommended) ............................................................... 10
  - Purge Objects for a Class (Use Extreme Caution!) .................................................. 11
  - Purge Objects (Do not use!) ................................................................................... 11
- **Using Workflow with WorkView** .......................................................................... 12
  - System Events ........................................................................................................... 12
- **Pop Integrations & URL Object Creation** ............................................................... 12
  - Viewing Objects ....................................................................................................... 12
  - Creating Objects ...................................................................................................... 13
  - Viewing Filter Results .............................................................................................. 13
Background
WorkView is a module that enables rapid development of applications suited for case management. These applications are based on relational database tables allowing for one-to-many or many-to-many relationships between objects where applicable. Since this is within the OnBase database, this also allows for integration with Unity forms and Workflow, and documents. WorkView applications are accessed through the Unity and web clients that users are already familiar with.

CU has 50 concurrent licenses, meaning 50 users can be using developed WorkView applications in the Unity/web clients at any time. Developers configuring in Studio do not count toward this limit.

Prerequisites
OnBase Studio must be installed to configure WorkView (and any related Workflow processes). OnBase Configuration is needed to create user groups (and any document types or keyword types related to your application).

Refer to the OnBase Client Guides for installation instructions.

Required Trainings
Prior to receiving access to configure WorkView, the following trainings must be completed:

- Work View | Case Manager Hands On Lab
- Work View | Case Manager: Building Your First Solution
- Work View | Case Manager: Intermediate
- Work View | Case Manager: Advanced
- Work View | Case Manager: Troubleshooting

Hyland also offers a WorkView Certification and hands-on sessions during the first two training days at CommunityLive. These are not strictly required but are strongly recommended due to the complexity and potential impact of WorkView configuration.

Resources
For more information, refer to:

- the WorkView MRG
- the Application Builder Community
- the WorkView | Case Manager Community

Additional recommended trainings:

- Workflow & WorkView: Building a Complete Solution
- WorkView: Calculated Attributes
• WorkView: Data Sets  
• WorkView: Notifications with ObjectPop & FilterPop  
• WorkView: Associations for Forms, Document Types & Folders  
• WorkView: Application Creation Excelerator (ACE)

If you need assistance from UIS, please contact UIS_DM_SUPPORT@cu.edu.

---

**User Groups & Access**

**Functional & WorkView User Groups**

Functional groups exist to support WorkView usage and development:

- **Functional - Level 4**: Provides basic user access to WorkView functionality in the Unity and web clients.
- **Functional - Developer - WorkView**: Required to configure WorkView in Studio (see Required Trainings)

No WorkView configuration (applications, classes, etc.) should be assigned to these functional groups; they exist only to provide access to WorkView functionality.

All other configuration should be assigned to “WorkView” user groups as outlined in Expectations and Tips for Certified Admins. No functional permissions should be assigned to these groups in Configuration, just create the group and add members. Then, assign only WorkView configuration (applications, classes, views, filter bars, etc.) in Studio.

**MANAGER Group**

Add the MANAGER group to all WorkView configuration items.

Do not enable the “Always On” option for the MANAGER group. This option can be selected for your other “WorkView” groups as needed.

**Master Security**

You can review a groups access to an application by right clicking the application and choosing “Master Security”. There you can copy access between groups, eliminate all access for a group, or grant access to an entire application for a group.

Master Security is the only place to review or modify access to Filter Bar Items after they are created. The groups shown at the Filter Bar level only refer to the bar, not the items assigned to the bar.
WorkView Configuration Guidelines

All WorkView configuration must conform with the OnBase Naming Conventions. Some items cannot include spaces or non-letter characters in their names, so the naming convention is modified accordingly.

All configuration must be developed in the DMOTST or DMOSTG environment and thoroughly tested prior to migration to DMOPRD.

WorkView requires careful consideration of the requirements and desired outcome before configuration.

Classes

1. It can be difficult to remove relationships between classes once they are established. Carefully plan which classes will be needed, how they might relate to each other, and what type of relationship(s) would exist (one to one, one to many, many to many) in order to configure the correct type of classes and relationships between them.

2. Do not configure Shared classes between applications.
   - Changes made to a shared class in one application also changes the class in all other applications using the class.
   - Shared classes also cause all other applications using the class to be included in any export files upon migration.
   - Password protection on an application affects other applications using the shared class.

3. To use existing employee and/or student data (from AutoFill Keyword Sets), configure External Classes as follows:
   - Create a new class and choose External Class.
   - Enter the class name.
   - Choose Local or Linked Server.
   - Set Server as local and the Database as the environment name (DMOPRD, DMOSTG, DMOTST or DMODEV).
     i. For student information, enter HSI.KEYSETDATA112 as the table name.
     ii. For employee information, enter HSI.KEYSETDATA117 as the table name.
• Assign the appropriate user groups (including MANAGER).
• For student information, configure attributes as follows. You can exclude any attributes you do not need but must include KEYSETNUM and Student ID.

<table>
<thead>
<tr>
<th>Name</th>
<th>Display Name</th>
<th>Data Type</th>
<th>Index</th>
<th>External Data Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYSETNUM</td>
<td>KEYSETNUM</td>
<td>Integer</td>
<td>Unique</td>
<td>KEYSETNUM</td>
</tr>
<tr>
<td>StudentID</td>
<td>Student ID</td>
<td>Alphanumeric (9)</td>
<td></td>
<td>KS113</td>
</tr>
<tr>
<td>FirstName</td>
<td>First Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS105</td>
</tr>
<tr>
<td>MiddleName</td>
<td>Middle Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS206</td>
</tr>
<tr>
<td>LastName</td>
<td>Last Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS104</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>Date of Birth</td>
<td>Date</td>
<td></td>
<td>KS106</td>
</tr>
</tbody>
</table>

• For employee information, configure attributes as follows. You can exclude any attributes you do not need but must include KEYSETNUM and Employee ID.

<table>
<thead>
<tr>
<th>Name</th>
<th>Display Name</th>
<th>Data Type</th>
<th>Index</th>
<th>External Data Column</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYSETNUM</td>
<td>KEYSETNUM</td>
<td>Integer</td>
<td>Unique</td>
<td>KEYSETNUM</td>
</tr>
<tr>
<td>EmployeeID</td>
<td>Employee ID</td>
<td>Alphanumeric (15)</td>
<td></td>
<td>KS102</td>
</tr>
<tr>
<td>FirstName</td>
<td>First Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS105</td>
</tr>
<tr>
<td>MiddleName</td>
<td>Middle Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS206</td>
</tr>
<tr>
<td>LastName</td>
<td>Last Name</td>
<td>Alphanumeric (50)</td>
<td></td>
<td>KS104</td>
</tr>
<tr>
<td>DateOfBirth</td>
<td>Date of Birth</td>
<td>Date</td>
<td></td>
<td>KS106</td>
</tr>
<tr>
<td>OPERID</td>
<td>Operator ID</td>
<td>Alphanumeric (10)</td>
<td></td>
<td>KS260</td>
</tr>
<tr>
<td>Email</td>
<td>Email</td>
<td>Alphanumeric (100)</td>
<td></td>
<td>KS529</td>
</tr>
<tr>
<td>Last4SSN</td>
<td>Last 4 SSN</td>
<td>Alphanumeric (4)</td>
<td></td>
<td>KS640</td>
</tr>
</tbody>
</table>

4. Extension Classes
• For sibling classes to share attributes, the attributes must be on the base class.
• For objects that will be converted to another class (ex. Base class = Person with extension classes of Applicant and Employee, and objects converted from Applicant to Employee), create relationships on the base class.

5. Association classes allow for many to many relationships.
   • Create the association class to relate to classes. Ex. CustomerXCase
   • Create a filter for each class related to the association class.
   • Embed each filter on a view for the opposite class. For example, put the customer filter on the case view and the case filter on the customer view.
   • Advanced associations allow for attributes to be added to the relationship, otherwise no attributes can be added to a standard association class. Advanced associations are created as a standard class but then configured with relationships to the two classes that should be related, along with the other applicable attributes.

6. External classes can be created to access data that is available in mviews from ICS, FIN and HCM. However, these external classes must be created using an ODBC connection and must be configured by UIS.
   • External classes also have some limitations, so please refer to the MRG. For example, values from external classes cannot be used as user prompts on a filter.

7. Disable direct object creation on classes where applicable (where users should never be able to create objects (ex. if objects will only be created by workflow) or where users should only create objects from within an embedded filter or through another defined process).

8. WorkView folders are easier to configure than File Cabinet Folders, but do not have a hierarchical structure. Either type of folder can be added to a view once it is configured, in order to show documents related to the object.

9. Limited actions are available directly within WorkView Actions.
   • For more advanced functionality:
     i. Configure actions to add objects to workflow. Once the object is in a queue, you can use the full set of rules and actions available there.
     ii. Configure actions to execute a System Task.
   • If an action is configured as a button on a view, it will no longer appear in the Tasks menu ribbon (it will only be available in one place).

10. Mappings/Associations.
    • More information: WorkView: Associations for Forms, Document Types & Folders and the Class Mappings section of the WorkView MRG.
• When configuring mapping for object creation or document import, map as many attributes as you can.
• When configuring mapping for searching for existing items, it’s better to use fewer attributes (since AND logic is used to identify items).
• Mapping/Association Types:
  i. Forms: Create forms based on data in the WorkView class
  ii. Document Types:
      1. Search for documents related to a WorkView object (including populating a WorkView folder)
      2. Create WorkView objects based on a document (from document retrieval or workflow)
  iii. Folder Types: Map WorkView objects to configured (filing cabinet) folder types to display the documents in the folder
  iv. Document Import: Import documents from WorkView and inherit values from the object to fill keyword values.
  v. Keyword Type Mappings: support the other mapping/association types

11. Create Mappings before creating Triggers.

12. Use Triggers when calculating values that will need to update or be used elsewhere. Calculated attributes are only evaluated once and are not as widely available for use within the application. See WorkView: Calculated Attributes for more details.

Filters and Filter Bars

1. Ensure all filters accessible to users in the clients have constraints or required prompts to limit the result list.

2. Naming filters with a prefix (along with the standard CU naming convention) makes it easier to identify where and how each filter should be used. It can also help to add the class name and the view name if applicable.
   • UI - User interface, likely used in filter bars
   • EMB - Embedded (within a view)
   • LU - Lookup (for in a view)
   • DS - Data set
   • WF - Used for Workflow

3. When assigning the filter to a filter bar as a filter bar item, it can be given a different display name.

4. Set a User Defined Filter Bar for the application to allow users to create their own filters.

5. There are a variety of macros available to make filters more complex/flexible, but note that the ~?FilterName option will not work if the filter used as criteria returns more than 2,000 results.
6. The “Enable Maximum Results” option on each filter overrides the global setting, but use caution and test as increasing the limit can result in performance issues.

7. Removing filter bar items requires a cache reset before users can access the app again. If a filter bar item is removed and cache is not reset, the application will fail to load in the client.

Views
1. Consider which fields should be required and/or read-only for your process.

2. Making the layout narrower makes it easier to read and use, a width of 85% works well.

3. Multiple views can be configured which will appear as tabs across the screen so that users do not need to scroll as much.

4. Any embedded items will run queries to load the items within, which can delay the load time. Embedded items such as folders may be better suited to their own views to improve load time.

5. Scripts are necessary to make views interactive (show/hide elements, make required/not required). In some cases (like if one group should be able to change an attribute value and another group should not), you may be able to create copies of the view and adjust the settings and assign to the appropriate groups instead of using scripts.

Application Creation Excelerator
This feature enables WorkView application creation based on a template created in MS Excel. This can reduce the time needed for development of applications.

Pre-defined ACE files are available on Community. A blank template can also be created from Studio.

Refer to Application Creation Excelerator in the WorkView MRG and WorkView: Application Creation Excelerator (ACE) for more details.

Testing
Resetting the app server cache is not sufficient for most WorkView changes to take effect in the Unity and/or web client; an app pool recycle is needed. However, recycling the app pool disrupts any other users in the environment, so it is necessary to wait until the next day to verify changes in the client. This includes, but is not limited to:

- Class creation/modification
- Permission modification
• Filter/Filter bar creation/modification

In general, plan to wait until the next day to test changes in WorkView.

---

**Importing Data to Classes**

Test data or pre-existing data can be imported from an Excel or CSV file using the WorkView menu in Studio. Refer to *Exporting and Importing Data* in the WorkView MRG for more information. This is separate to any export/import of configuration.

Item generator can also be used to create objects or documents used to create objects for testing purposes. Refer to *Configuring Item Generators* in the Studio MRG for more information.

---

**Migration**

Refer to the *Exporting and Importing Configurations* guide for details on migration. There are special considerations for migrating projects involving WorkView, especially the reporting elements.

Note that the Data Import/Export options on the WorkView menu in Studio cannot be used for WorkView configuration migration; this only imports or exports data for objects.

---

**Purging Objects**

Similar to how deleted documents can be recovered until they are purged from Document Maintenance, deleted objects still exist in the database (but are not visible to users in the client) until they are purged. There are a few different methods of purging objects, but all are **permanent and irreversible** so it is essential that you carefully review and confirm you are taking the right action.

Always confirm which environment/repository you are working in. To avoid any potential mix-ups, it is recommended to only be connected to one environment in Studio when performing these actions.

If you have any doubt, please contact UIS_DM_Support@cu.edu for assistance before proceeding.

**Purge Objects Selectively (Recommended)**

This menu option (WorkView tab in Studio) is the best way to purge objects.
A new window will open where you can select the desired application, class and filter.

By default, filters will only show active objects, so if you want to purge objects that are already deleted, you will need a filter that is configured to show deleted objects. This is an option available on the Advanced tab in the filter configuration.

<table>
<thead>
<tr>
<th>General</th>
<th>Advanced</th>
<th>Filter Bar Items</th>
<th>Events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display as Class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;Default&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Return Distinct Items</td>
<td>Group Like Constraints</td>
<td>Ability to Select Multiple Rows</td>
<td>Show Entire Text Values</td>
</tr>
<tr>
<td>Show Only Deleted Objects in Filter</td>
<td></td>
<td></td>
<td>Display as Dashboard</td>
</tr>
</tbody>
</table>

**Purge Objects for a Class (Use Extreme Caution!)**

After right clicking a class, there is an option in that menu to Purge Objects. This will purge ALL objects in that class, regardless of whether they are active or deleted. After doing so, no data will exist in the class.

There are cases where this may be necessary, especially while testing in non-production. Carefully review what environment you are in and the class name to ensure the right selection is made.

**Purge Objects (Do not use!)**

This menu option (WorkView tab in Studio) will purge ALL (regardless of application/class) inactive and deleted objects in the environment. Inactive objects are objects that were created, but were never saved. Since this will affect all applications and not just your own, do not use this option.
Using Workflow with WorkView
In order to route WorkView objects through a life cycle, the life cycle will need to be configured as (or converted to) a Unity life cycle. You can then assign application classes to the life cycle.

You can also identify objects as related items when processing documents in Workflow.

System Events
You can also use System Events with WorkView to enhance your applications with workflow functionality. System Events are similar to System Tasks but are triggered by the specified event (ex. when an object is saved) occurring rather than a user performing the task.

More information on System Events is in the Workflow MRG (page 246 for EP3). Make sure you reference the documentation for particular rules/actions since some are only actually available in certain contexts but will be available to select regardless such that you configure them where they won’t actually work.

Pop Integrations & URL Object Creation
Pop integrations allow for interaction with OnBase content through URLs. More information on the following is available in the WorkView: Notifications with ObjectPop & FilterPop course and in the following sections of the WorkView MRG:
- Using ObjectPop in Notifications (page 295 for EP3)
- Using UnityPop in Notifications (page 297 for EP3)
- Creating Objects Using ObjectPop (page 296 for EP3)
- Creating Objects Using a URL (page 396 for EP3)
- FilterPop (page 806 for EP3)
- Displaying Filter URLs (page 397 for EP3)

For any pop integration URLs, use the corresponding values for the environment:

<table>
<thead>
<tr>
<th>SERVER</th>
<th>VIRTUAL_ROOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV dm-dev.dev.cu.edu</td>
<td>AppNet</td>
</tr>
<tr>
<td>TST dm-tst.qa.cu.edu</td>
<td>AppNet</td>
</tr>
<tr>
<td>STG dm-stg.qa.cu.edu</td>
<td>AppNet</td>
</tr>
<tr>
<td>PRD dm.prod.cu.edu</td>
<td>AppNet</td>
</tr>
</tbody>
</table>

For any of these actions, authentication to an individual OnBase account with sufficient privileges is required.

Viewing Objects
ObjectPop can be used to direct users (with a URL) to open a WorkView object.
http://<Server>/AppNet/workview/objectPop.aspx?objectid=<object ID>&classid=<class ID>

This can also be done with a UnityPop URL to open an existing object in the Unity client. To provide a static link to a specific object, the following format should be followed:

onbase://wv/object/?classid=<class ID>&objectid=<object ID>

**Creating Objects**

Objects can also be created using a URL. The ctype property defines whether the object is active or inactive (0 active; 1 inactive). You can also set attribute values on the object that is created.

**ObjectPop Object Creation Example URL:**

http://<Server>/AppNet/workview/objectPop.aspx?classid=<class ID>&ctype=<0 or 1>&<attribute ID>=<attribute value>

**Object Creation UnityPop Example URLs:**

onbase://wv/create/object/?classid=1001&ctype=0
onbase://wv/create/object/?classid=1001&1001=Open&ctype=0
onbase://wv/create/object/?classid=1001&Status=Open&ctype=0
onbase://wv/create/object/?classid=1001&1002.1003=John%20Adams&ctype=1

**Viewing Filter Results**

FilterPop can be used to direct users (with a URL) to a WorkView filter. A tool exists to help with creation of the FilterPop URL:

https://<Server>/AppNet/workview/filterPop.aspx

You will end up with a URL such as:

https://dm.prod.cu.edu/AppNet/Workview/filterPop.aspx?filterPop=true&filterID=XXXX&Keywords=XXXX_%3D_AND_OPEN

This can also be done with a UnityPop URL to view a filter's results in the Unity client. To provide a static link to a filter, the following format should be followed:

onbase://wv/filter/?id=1001

or, with constraints

onbase://wv/filter/?id=1001&constraints=1001.1002=_and__value