

# Workflow Development Best Practices

Last Modified on 01/16/2018 7:06 am EST

Workflows are the core of business process management. It is important that you develop workflows according to best practices.

## Workflow Composition

Workflow visualization affects and aids the collaboration on workflows between cross-functional teams.

- Develop an effective naming convention.
  - Use meaningful names
  - Use understandable names
  - Be consistent with names
- When possible, add swimlanes to divide the workflow. Swimlanes are intended to divide the workflow by user or department. Swimlanes help optimize workflows, because you can easily identify areas with too many activities.
- Add comments and labels to make the workflow easier to understand (especially for business users).
- Do not add too many empty activities for visual effect. This can cause inflated DB tables with waste data, and performance issues. Only use empty activities when necessary, for example, before split activities, and to ensure clear layout.

## Forms Development

### Data Model

- There are two Data Model methodologies
  - Create the data model, then view
  - View first
- If you create the data model first, plan your data model during the requirements stage.
- Use appropriate and meaningful names
- Select appropriate types for the input you are expecting.
- We recommend using a primary key that consists of a single field.
- When selecting a key, remember:
  - For activity tables (UACTs), only use system keys
  - Select the system key according to your solution requirements (*fldwfID*, *fldiActID*, *fldAlid*)
- We recommend that you define the display field yourself, and not use the default, since it appears in several places.
- Use caution if you decide to share activity tables between workflows. When you share tables between workflows, the data is merged into a single table. This can cause the table to grow in size, unnecessarily.
- Do not place too many activity views on the same UACT table. This can cause the table to grow in size, unnecessarily. Each field in each table adds a column to the table, and combo boxes add two columns.
- Tables should contain a maximum of 40-50 columns.
- Consider whether a field should be nullable. If the field is not nullable, and the user does not enter a value, an error occurs. On the other hand, if the field is nullable, it ensures that the field is populated. To prevent an error, you can place a validator on this field in the view.

### Lookup Tables

- Lookup tables do not support multiple keys.
- Keep the names that you selected during the design. Since the lookup table is accessed by a path, if you change names in the lookup table, the reference breaks.

### Sub-Views

- Use sub-views to create reusable components, or to logically divide the form.
- Use caution when you use a sub-view from a different workflow. The sub-view's data model is saved with the other workflow. Therefore, if you need to export the workflow, you must export the other workflow.
- We recommend not changing the names of workflows, views, and activities because the path to the sub-view might cause errors. Spend time during the design stage to select appropriate workflow names.
- Consider using sub-views for header records, such as customer details, which remain consistent throughout the workflow. Since the view is maintained in a single place, you can accelerate process development and make maintenance easier.
- Do not embed sub-views within sub-views, within sub-views. Do not make the solution overly complex to understand, investigate, and troubleshoot.

### Workflow Variables

- Consider using workflow variables to expose key parameters, or to control logic, so they are simple to maintain.

### Look and feel

Follow these guidelines when you use a predefined color scheme and template

- Use CSS classes
- Use real SRC for images
- Do not use style property
- Do not use images from the attachments of any other dynamic location

### Creating Reusable Content

Cora SeQUENCE provides several opportunities for creating reusable components. Take advantage of reusable content to save time and effort.

- Sub-views
- Sub-workflows
- Global variables
- Extending form templates

**Tip:** Create a general workflow that includes a list of activities that represent various types of controls that you want to reuse.