Acceleration Framework Extensions (AFE) Deployment Guide

Ad-Hoc Workflow v1.0

[Requirements](#_6ixh1w6tjnrv)

[Required Environment Configurations (maintenance window and downtime required)](#_avsl7olceuvp)

[Required Plugins](#_bsd4lxf7lk8y)

[Required Application Packages](#_gy0bgg35xkh4)

[Required Database Scripts](#_yktcwxsvxjzz)

[Required Users](#_ae5lc1dhrpl0)

[Required Images](#_dtaayo89dgzf)

[Configuration](#_ow3onuua7npr)

[Revert Changes (Back Out Plan)](#_xnvjqpuxcnkn)

# Requirements

The Ad-Hoc Workflow Acceleration Framework Extension is configured for Appian Software Version 16.2+ and MySQL Database

## Required Environment Configurations (maintenance window and downtime required)

* None

## Required Plugins

* None

## Required Application Packages

* Acceleration Framework Extensions - Ad-Hoc Workflow 1.0.zip

## Required Database Scripts

* MySQL
  + AFE Ad-Hoc Workflow Release 1.0 Create DB.sql
  + AFE Ad-Hoc Workflow Release 1.0 Insert Reference Data.sql

## Required Users

* None

## Required Images

* None

# Configuration

1. Confirm target environment already has **AF Common Objects 2.0 – All Objects.zip** deployed. If not, refer to the [Acceleration Framework Extensions Deployment Guide - Baseline](https://docs.google.com/a/appian.com/document/d/1IGYeZwAcGmQpMv_NhmHvE7lne4_eQbhCaiRs70un1W0/edit?usp=sharing) before moving on to next step
2. Confirm target environment already has **Acceleration Framework Extensions Baseline.zip** deployed. If not, refer to the [Acceleration Framework Extensions Deployment Guide - Baseline](https://docs.google.com/a/appian.com/document/d/1IGYeZwAcGmQpMv_NhmHvE7lne4_eQbhCaiRs70un1W0/edit?usp=sharing) before moving on to next step
3. Deploy Ad-Hoc Workflow 1.0 Acceleration Framework Extension
   1. Create table structure utilizing the database appropriate scripts (in the following order):
      1. For MySQL database:
         1. **AFE Ad-Hoc Workflow Release 1.0 Create DB.sql**
         2. **AFE Ad-Hoc Workflow Release 1.0 Insert Reference Data.sql**
      2. For non-MySQL databases:
         1. Convert the following scripts to SQL Server syntax, Oracle syntax, etc. and run in the following order:
            1. **AFE Ad-Hoc Workflow Release 1.0 Create DB.sql**
            2. **AFE Ad-Hoc Workflow Release 1.0 Insert Reference Data.sql**
   2. Import **Acceleration Framework Extensions - Ad-Hoc Workflow 1.0.zip**
4. Manually set the appropriate values for the environment specific values that will be applied to all workflow, regardless of associated record type:
   1. Navigate to Rules > AFE > Utility
      1. Set **AFE\_RULE\_Utility\_Workflow\_setRelatedActionExpiration** = <<Expiration time to be specified by deployment manager>>
         1. By default, set to 20 minutes
   2. Navigate to Rules > AFE > Rules
      1. Set **AFE\_RULE\_Workflow\_setTaskReminderDatetime** = <<Reminder time to be specified by deployment manager>>
         1. By default, set to 24 hours
   3. Navigate to Rules > AFE > Display
      1. Set **AFE\_RULE\_Workflow\_displayTaskReminderAlert** = <<Exact wording to be specified by deployment manager>>
         1. By default, set to “This task is due on <<deadline>>”
      2. Set **AFE\_RULE\_Workflow\_displayTaskReminderSubject** = <<Exact wording to be specified by deployment manager>>
         1. By default, set to “Reminder: <<Task Name>>: <<Workflow Name>>”
5. Enable Ad-Hoc Workflow for a specific target record type. **(NOTE: The target record type must have an integer-type primary key)**
   1. Insert target record type into database reference table:
      1. Copy INSERT statement:
         1. INSERT INTO `AF\_R\_DATA` (`R\_TYPE`, `R\_LABEL`, `R\_IS\_ACTIVE`) VALUES ('RECORD\_TYPE', 'Target Record Type Name', 1);
      2. Replace ‘Target Record Type Name’ with actual target record type’s singular name
      3. Execute the INSERT statement
      4. Note the corresponding auto-generated R\_ID primary key for the target record type
         1. (Optional) Create an environment specific constant to store this value per implementation/project-specific guidelines for managing environment specific constants to be referenced in the steps below instead of <<Target Record Type R\_ID primary Key>>
   2. Update logic in **AFE\_RULE\_Workflow\_returnRecordType** to properly return the target recordType object
      1. Navigate to Appian Design interface
      2. Click on Objects
      3. Search for **AFE\_RULE\_Workflow\_returnRecordType**
      4. Open and edit the expression rule to add a new (nested, if at least one other record type already exists for ad-hoc workflow) if() condition for the target record type R\_ID primary key
      5. Save changes to the expression rule
   3. Update logic in **AFE\_RULE\_Workflow\_returnGroupFilter** to properly return the target group of users to be selectable as assignees and/or initiators for any workflows for the target record type
      1. Navigate to Appian Design interface
      2. Click on Objects
      3. Search for **AFE\_RULE\_Workflow\_returnGroupFilter**
      4. Open and edit the expression rule to add a new (nested, if at least one other record type already exists for ad-hoc workflow) if() condition for the target record type’s group of valid assignees
      5. Save changes to the expression rule
   4. Configure user security for permissions/access to ad-hoc workflows
      1. Add same target group of users to be selectable as assignees and/or initiators for any workflows for the target record type to parent group **AFE Ad-Hoc Workflow All Users** and parent group **AF All Users**
      2. Add target group of Appian Administrators for any workflows for the target record type to parent group **AFE Ad-Hoc Workflow Appian Administrators**  and parent group **AF Appian Administrators**
   5. Add related action to kick off an ad-hoc workflow to target record type
      1. Navigate to target record type
      2. Open and edit the record type
         1. Under **Related Actions**, click **+ New Related Action**
            1. Process Model: **Take Action on This Record**
            2. Context:

**{**

**recordId: rp!id,**

**recordTypeId:** <<Target Record Type R\_ID primary Key>>

**}**

* + - * 1. Visibility: <<Configure as needed per security requirements>>
        2. Click OK
    1. Save changes to the record type
  1. (Optional) Add record view to display all created, in progress, and completed ad-hoc workflows
     1. Navigate to target record type
     2. Open and edit the record type
        1. Under **Views**, click **+ New View**
           1. View Name: <<Record View Name>>

Recommended View Name: =“Workflows”

* + - * 1. Interface:

**=rule!AFE\_DASH\_Workflow\_workflow(**

**recordTypeId:** <<Target Record Type R\_ID primary key>>**,**

**recordId: rp!id**

**)**

* + - * 1. Visibility: <<Configure as needed per security requirements>>
    1. Save changes to the record type
  1. (Optional) Display all current tasks/assignments from all in progress workflows on a Record View
     1. Navigate to target record view interface
        1. Recommended record view: Summary
     2. Open and edit the interface
     3. Add two new singular parameters:
        1. For record type
           1. Name: **recordTypeId**
           2. Type: **Number (Integer)**
        2. For specific record ID
           1. Name: **recordId**
           2. Type: **Number (Integer)**
     4. Add a new section to display the current tasks/assignments:

**rule!AFE\_SCT\_Workflow\_tasks(**

**recordTypeId: ri!recordTypeId,**

**recordId: ri!recordId**

**)**

* + 1. Save changes to the record view interface
    2. Navigate to the target record type
    3. Open and edit the record type
       1. Under **Views**, open and edit the target record view
          1. For Interface, pass in values for new parameters

**recordTypeId:** <<Target Record Type R\_ID primary key>>

**recordId: rp!id**

* + 1. Save changes to the record type

# Revert Changes (Back Out Plan)

The following steps will be taken if the deployment into the target environment needs to be reverted:

* Notify stakeholder of the deployment rollback
* For Cloud environments: Snapshots are automatically taken on a daily basis. The deployment team will need to coordinate with Appian Support to revert to the last snapshot.
  1. Create an Appian Support Case with instructions to revert back to a prior snapshot. Include the following information:
     + Target environment to revert
     + Snapshot date to revert to
* For On-Premise environments: Immediately prior to beginning deployment, create a backup of the existing Appian installation. Revert to this backup if the deployment needs to be rolled back.
  1. Properly shutdown Appian
     + Ensure the engines are checkpointed upon shutdown
     + Ensure any scheduled processes are disabled during the deployment window
  2. Backup the Appian installation on all web servers, app servers, and engine servers
     + Create a copy of the entire <<APPIAN\_INSTALL>> directory of your Appian instance and rename it to clearly denot it as a backup copy (e.g. “<<APPIAN\_INSTALL>>\_<<CURRENT\_DATETIME>>\_bkup”)
     + Backup the Apache directory on the web server(s)
     + Backup any fileshare documents (for multiple JBoss instances sharing a fileshare). If not using fileshare, backup the \_admin and other data directories
  3. Backup the primary and business databases
  4. Properly start Appian and begin deployment procedure outlined above
  5. If necessary to roll back the deployment and revert any changes, properly shutdown Appian
  6. Rename the entire <<APPIAN\_INSTALL>> directory of your Appian instance on all web servers, app servers, and engine servers to clearly denote it as a failed attempted deployment copy (e.g. “<<APPIAN\_INSTALL>>\_<<CURRENT\_DATETIME>>\_failed”)
  7. Rename the backup copy to the original <<APPIAN\_INSTALL>> directory name on all web servers, app servers, and engine servers
  8. If using a fileshare for multiple JBoss servers, replace the contents with the backup taken in the previous step
  9. Restore the Appian primary database with the previous backup taken
  10. Restore the Appian business database with the previous backup taken