

Search documentation

Search

Appian

Standard Object Names

Appian 7.7

NOTE: This site does not include documentation on the latest release of Appian. Please upgrade to benefit from our newest features. For more information on the latest release of Appian, please see the [Appian 7.8 documentation](#)

We recommend the following naming conventions for Appian objects. Adapt these standards to your environment as needed.

Toggle Na

Appian Object	Recommended Naming Standard
Custom Data Types	<ul style="list-style-type: none"> Provide descriptive names for your custom data types (CDTs). Use a custom namespace for your CDT to group them and associate them with a specific application. Changing the namespace will require editing the XSD for the CDT.
Variables	<p>Use <code>camelCase_dataTypeAbbreviation</code> for defining variables. For example:</p> <ul style="list-style-type: none"> <code>firstName_txt</code> <code>salaryAdvanceAmount_dec</code> <code>generalExpenseList_cdt</code> <p>Using <code>camelCase</code> helps differentiate words without adding clumsy underscores. Using a lowercase <code>dataTypeAbbreviation</code> helps clarify the variable <code>types</code> without requiring that the user reference the process model properties or process documentation.</p>
Rules and Constants	<p>For Rules and Constants, use an application abbreviation prefix and optionally an additional prefix to group related Rules and Constants. Also, Constants should be all uppercase. For example:</p> <ul style="list-style-type: none"> <code>HR_ComputeBaseSalary()</code> <code>HR_IMG_CAREER_HISTORY_ICON</code> <code>HR_TempoMessage_NewEmployee</code> - this is a Rule not a Constant because it generates a dynamic message based on the employee's name and start date. <p>Also, the Create Skeleton Application-Specific Hierarchies page explains how to setup your hierarchy of Rules and Constants folders.</p>
Groups	<ul style="list-style-type: none"> Create a parent group named for each application. Name child groups according to their security role. Each application should include the following child groups at a minimum. <ul style="list-style-type: none"> Application Administrators Application Users
Process Models	<p>Preface your process model and process model folder names with the name or acronym of the associated application. For example, a Customer Knowledge Base application's bulk upload process model would be named CKB Bulk Upload.</p> <p>The Create Skeleton Application-Specific Hierarchies page explains how to setup your hierarchy of process model folders.</p>
Knowledge Centers and Communities	<p>Preface your knowledge center names with the name or acronym of the associated application. For example, a Customer Knowledge Base application's document management knowledge center might be named CustomerKB KC. Use a single community named for the application, such as CustomerKB.</p> <p>The Create Skeleton Application-Specific Hierarchies page explains how to setup your hierarchy of document folders.</p>
Reports	<p>Store reports in the application community. Apply the appropriate security so that users can see the reports.</p> <p>The Create Skeleton Application-Specific Hierarchies page explains how to setup your hierarchy of Rule folders.</p>
Images	<p>For displaying images, do not link to images on the application server (such as <code>/suite/bank/bank_picture.jpg</code>). Upload all images to the application community in Appian Document Management and apply the appropriate security so that all users can see that image. Then create a Constant to reference that image. Then use that Constant in your Web Content with Dynamic Details channel or in your Form.</p>

User Guides by Role

Designer

Developer

Web Admin

Server Admin (On-Premise Only)

Tutorials

Records

Interfaces

Process

Release Information

Release Notes

Installation

Migration

System Requirements

Hotfixes

Release History

Other

STAR Methodology

Best Practices

Glossary

APIs
