

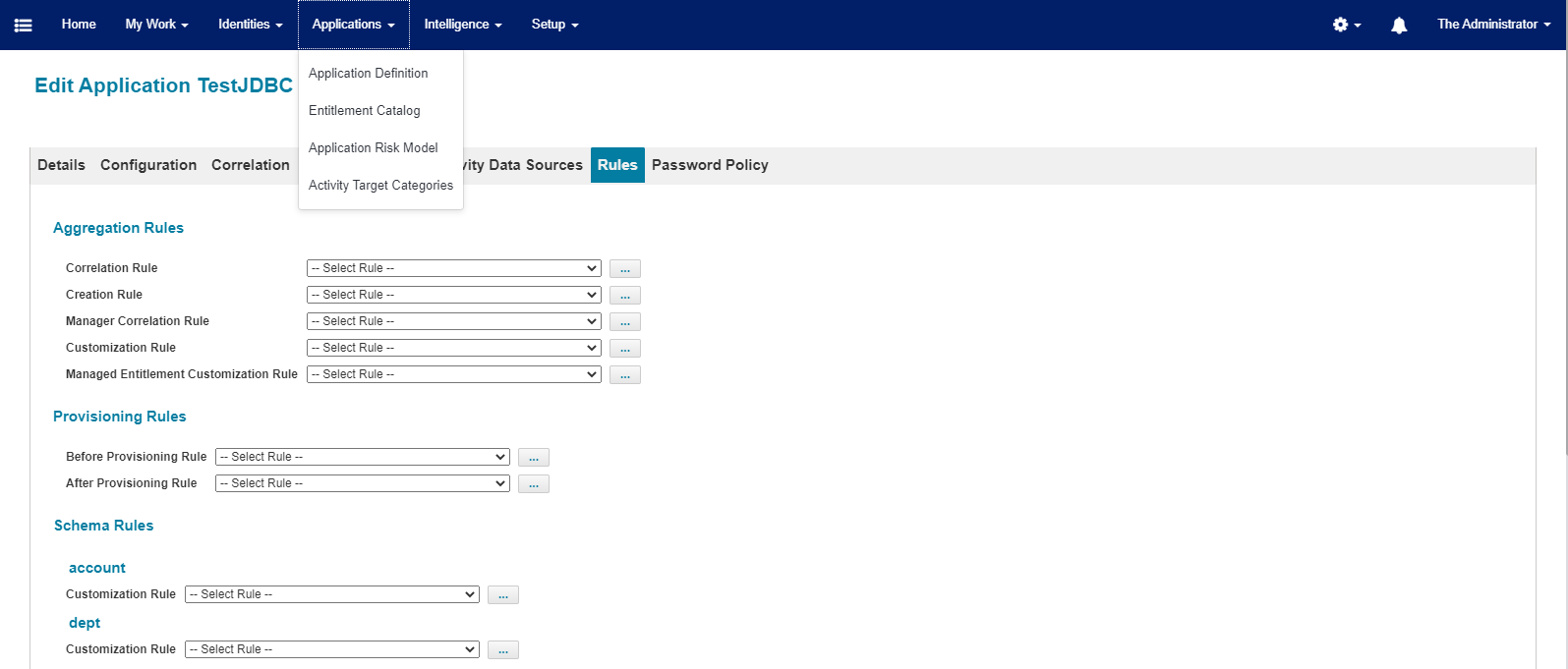
This department as manged attributes are populated in entitlement catalogue after doing Aggregation.

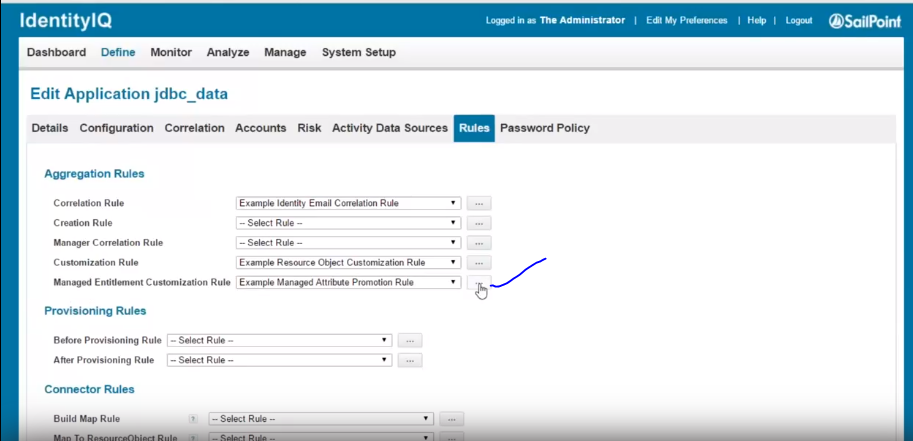
These are Non-requestable by default.

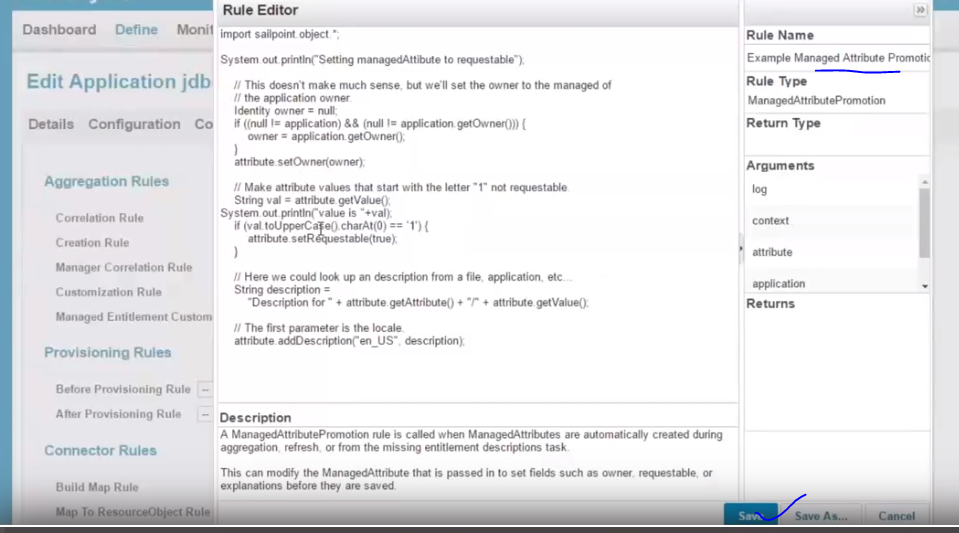
You cannot request for access for other

These roles cannot be requested any one from this section.

You can make requestable **go to->Application->Application Definition->Rules**







Now the above entitlements are requestable whose value =1. It means it is requestable by using the **Managed Entitlement customisation Rule**.

#

# (c) Copyright 2018 SailPoint Technologies, Inc., All Rights Reserved.

#

# NOTE: The syntax of this file is different than what it was for

# IdentityIQ 7.3 and earlier. These syntax changes are due to

# our upgrade to log4j version 2.

#

# See https://logging.apache.org/log4j/2.x/manual/configuration.html

# for description of the new syntax.

#

######################################

## Global log4j2 properties

######################################

name=identityiq\_default

status=warn

monitorInterval=20

packages=sailpoint.api.logging

#######################################

## Appenders

#######################################

### direct log messages to stdout ###

appender.stdout.type=Console

appender.stdout.name=stdout

appender.stdout.target=SYSTEM\_OUT

appender.stdout.layout.type=PatternLayout

appender.stdout.layout.pattern=%d{ISO8601} %5p %t %c{4}:%L - %m%n

# CRLF log injection can reduce the value of application logs. CRLF log injection can

# be neutralized by applying log4j2's replace conversion pattern to a log message. If

# log4j2 environment is set up to use the PatternLayout, then while in stdout or file

# write, an administrator can use the 'replace' conversion pattern to a log message to

# neutralize CRLF sequence to double underscore. In the example below, the inner replace

# pattern replaces a sequence of line feed (\n) and carriage return (\r) to two underscores

# respectively, whereas the outer replace pattern replaces the carriage return and line feed

# sequence of a log message to two underscores respectively. Adjust double underscore from

# both the places as per one's own choice of sequence.

#appender.stdout.layout.pattern=%d{ISO8601} %5p %t %c{4}:%L - %replace{%replace{%m}{\n\r}{\_\_}}{\r\n}{\_\_}%n

### direct log messages to the db

appender.syslog.type=spsyslog

appender.syslog.name=spsyslog

# Below is an example of how to create a logger that writes to a file.

# Uncomment the following five lines, then uncomment the

# rootLogger.appenderRef.file.ref definition below

#appender.file.type=File

#appender.file.name=file

#appender.file.fileName=C:/Windows/Temp/sailpoint.log

#appender.file.layout.type=PatternLayout

#appender.file.layout.pattern=%d{ISO8601} %5p %t %c{4}:%L - %m%n

#

# SailPoint metering is useful to help diagnose performance issues.

# Some critical sections of code will gather performance data

# metrics, and will log the data to the meter appender.

#

# Below is an example of how to create a csv of metered data.

# Uncomment the following 11 lines, as well as the logger.apiMeter lines

# near the end of this file

#

#appender.meter.type=RollingFile

#appender.meter.name=meter

#appender.meter.fileName=C:/Windows/Temp/meter.log

#appender.meter.filePattern=C:/Windows/Temp/meter-%d{yyyy-MM-dd}-%i.log.gz"

#appender.meter.layout.type=PatternLayout

#appender.meter.layout.pattern=%m%n

#appender.meter.policies.type=Policies

#appender.meter.policies.size.type=SizeBasedTriggeringPolicy

#appender.meter.policies.size.size=10MB

#appender.meter.strategy.type=DefaultRolloverStrategy

#appender.meter.strategy.max=5

#######################################

## rootLogger

#######################################

### set default log levels and appenderRef

### valid log levels in increasing order of severity are:

### trace, debug, info, warn, error, fatal, off

### trace is required to get method entry and exit logging

rootLogger.level=warn

rootLogger.appenderRef.stdout.ref=stdout

# Uncomment to also write to file appender by default.

# Also need to uncomment the file appender definition above.

#rootLogger.appenderRef.file.ref=file

#######################################

## 3rd-party loggers (recommended)

#######################################

# Suppress a warning about using a default configuration

logger.ehcache.name=net.sf.ehcache

logger.ehcache.level=error

# Suppress "narrowing proxy" hibernate warning.

logger.ohe\_spc.name=org.hibernate.engine.StatefulPersistenceContext

logger.ohe\_spc.level=error

# Suppress warnings about deprecated gadgets during import

logger.jasper\_engine\_xml.name=net.sf.jasperreports.engine.xml

logger.jasper\_engine\_xml.level=error

# Note: The following logging properties turn off warnings that result from our

# combination of technologies (Tomahawk/Faceletes/A4J, etc.).

# Commenting them out will result in a slew of parser warnings.

# If you don't want to clutter your logs with a bunch of useless warning

# messages please leave these as they are.

# Suppress warning message about Unparsable lastModified

logger.renderkit1.name=org.apache.myfaces.renderkit.html.util.MyFacesResourceLoader

logger.renderkit1.level=error

# Suppress warning message about response not having <head> or <body> tags

logger.renderkit2.name=org.apache.myfaces.renderkit.html.util.DefaultAddResource

logger.renderkit2.level=error

# Suppress warning message about invalid HTML inside javascript

# See http://www.mail-archive.com/users@myfaces.apache.org/msg50391.html

logger.renderkit3.name=org.apache.myfaces.renderkit.html.util.ReducedHTMLParser

logger.renderkit3.level=error

#######################################

## SailPoint loggers (required)

#######################################

# make sure the syslog code doesn't try to call itself if there's a problem.

# NOTE: The additivity flag is critical here!

logger.syslogAppender.name=sailpoint.api.logging.SyslogAppender

logger.syslogAppender.level=error

logger.syslogAppender.appenderRef.stdout.ref=stdout

logger.syslogAppender.additivity=false

logger.syslogEvent.name=sailpoint.object.SyslogEvent

logger.syslogEvent.level=error

logger.syslogEvent.appenderRef.stdout.ref=stdout

logger.syslogEvent.additivity=false

logger.sailpoint.name=sailpoint

logger.sailpoint.level=warn

logger.sailpoint.appenderRef.syslog.ref=spsyslog

# Hibernate logs a warn every time Criteria is used. Supress this until JPA migration is completed -rap

logger.hibernateDeprecation.name=org.hibernate.orm.deprecation

logger.hibernateDeprecation.level=error

#######################################

## SailPoint loggers (troubleshooting)

#######################################

#logger.orgHibernate.name=org.hibernate

#logger.orgHibernate.level=info

#logger.orgHibernate.level=debug

#logger.orgHibernate.level=trace

#logger.hibernateCache.name=org.hibernate.cache

#logger.hibernateCache.level=trace

#logger.orgSpringframework.name=org.springframework

#logger.orgSpringframework.level=info

#logger.orgQuartz.name=org.quartz

#logger.orgQuartz.level=info

# Uncomment to log all SQL and prepared statement parameter values.

#logger.hibernateSQL.name=org.hibernate.SQL

#logger.hibernateSQL.level=debug

#logger.hibernateType.name=org.hibernate.type

#logger.hibernateType.level=trace

# Uncomment this to log all SailPoint generated HQL queries and

# query parameters.

#logger.persistenceHQL.name=sailpoint.persistence.hql

#logger.persistenceHQL.level=trace

# Uncomment to view JSF logging

#logger.phaseTracker.name=sailpoint.web.util.PhaseTracker

#logger.phaseTracker.level=trace

#logger.sunFaces.name=com.sun.faces

#logger.sunFaces.level=trace

#logger.javaxFaces.name=javax.faces

#logger.javaxFaces.level=trace

# Uncomment to enable logging of performance metering data

#logger.apiMeter.name=sailpoint.api.Meter

#logger.apiMeter.level=info

#logger.apiMeter.appenderRef.meter.ref=meter

# Uncomment to enable authentication related logging

#logger.authFilter.name=sailpoint.web.PageAuthenticationFilter

#logger.authFilter.level=debug

#logger.authService.name=sailpoint.service.PageAuthenticationService

#logger.authService.level=debug

#logger.ssoValidator.name=sailpoint.web.sso.DefaultSSOValidator

#logger.ssoValidator.level=debug

#logger.ssoDefault.name=sailpoint.web.sso.DefaultSSOAuthenticator

#logger.ssoDefault.level=debug

#logger.ssoSAML.name=sailpoint.web.sso.SAMLSSOAuthenticator

#logger.ssoSAML.level=debug

# Uncomment to enable detailed multi-factor authentication logging

#logger.mfaFilter.name=sailpoint.web.MFAFilter

#logger.mfaFilter.level=debug

#logger.mfaLibrary.name=sailpoint.workflow.MFALibrary

#logger.mfaLibrary.level=info

# Uncomment to enable detailed statistics monitoring logging

#logger.monitoringService.name=sailpoint.server.MonitoringService

#logger.monitoringService.level=debug

#logger.aggregator.name=sailpoint.api.Aggregator

#logger.aggregator.level=trace

#logger.cacheTracker.name=sailpoint.api.CacheTracker

#logger.cacheTracker.level=trace

#logger.certificationer.name=sailpoint.api.Certificationer

#logger.certificationer.level=info

#logger.remediationManager.name=sailpoint.api.certification.RemediationManager

#logger.remediationManager.level=info

#logger.certificationPhaser.name=sailpoint.api.CertificationPhaser

#logger.certificationPhaser.level=info

#logger.correlationModel.name=sailpoint.api.CorrelationModel

#logger.correlationModel.level=info

#logger.lockTracer.name=sailpoint.api.LockTracker

#logger.lockTracer.level=trace

#logger.managedAttributer.name=sailpoint.api.ManagedAttributer

#logger.managedAttributer.level=info

#logger.provisioner.name=sailpoint.api.Provisioner

#logger.provisioner.level=info

#logger.oimClient.name=sailpoint.integration.oim.OIMClient

#logger.oimClient.level=info

#logger.objectConfig.name=sailpoint.object.ObjectConfig

#logger.objectConfig.level=info

#logger.dateType.name=sailpoint.persistence.DateType

#logger.dateType.level=info

#logger.debugInterceptor.name=sailpoint.persistence.DebugInterceptor

#logger.debugInterceptor.level=debug

#logger.hibernatePersistenceManager.name=sailpoint.persistence.HibernatePersistenceManager

#logger.hibernatePersistenceManager.level=trace

#logger.hibernatePersistenceManager.level=info

#logger.newXmlType.name=sailpoint.persistence.NewXmlType

#logger.newXmlType.level=info

#logger.sailPointInterceptor.name=sailpoint.persistence.SailPointInterceptor

#logger.sailPointInterceptor.level=info

#logger.xmlType.name=sailpoint.persistence.XmlType

#logger.xmlType.level=info

#logger.sailpointReporting.name=sailpoint.reporting

#logger.sailpointReporting.level=trace

#logger.bsfRuleRunner.name=sailpoint.server.BSFRuleRunner

#logger.bsfRuleRunner.level=trace

#logger.cacheService.name=sailpoint.server.CacheService

#logger.cacheService.level=info

#logger.serverEnvironment.name=sailpoint.server.Environment

#logger.serverEnvironment.level=trace

#logger.iiqRestService.name=sailpoint.service.IIQRestService

#logger.iiqRestService.level=info

#logger.idRefreshExecutor.name=sailpoint.task.IdentityRefreshExecutor

#logger.idRefreshExecutor.level=trace

#logger.roleSynchronizer.name=sailpoint.task.RoleSynchronizer

#logger.roleSynchronizer.level=info

#logger.timingFilter.name=sailpoint.web.util.TimingFilter

#logger.timingFilter.level=trace

#logger.heartbeat.name=sailpoint.server.HeartbeatService

#logger.heartbeat.level=trace

#logger.identityai.name=sailpoint.identityai

#logger.identityai.level=debug

#logger.connector\_sm.name=sailpoint.connector.sm

#logger.connector\_sm.level=debug

Enabling logs in