# SailPoint IdentityIQ Accessio RACF Connector

## Overview

The SailPoint IdentityIQ Accessio RACF Connector provides comprehensive integration between SailPoint IdentityIQ and the Garancy RACF system. This connector supports full user lifecycle management, role-based access control, multi-level approval workflows, segregation of duties (SoD) validation, compliance recertification, and comprehensive audit logging.

## Features

### Core Functionality

* **User Lifecycle Management**: Create, modify, delete, enable, and disable users
* **Role Management**: Assign and remove roles with SoD validation
* **Multi-Level Approval Workflows**: Support for AARID, STC, and Technical user types
* **Compliance Recertification**: Automated recertification campaigns
* **Audit Logging**: Comprehensive audit trail for all operations
* **Real-time Integration**: SOAP/XML API integration with Garancy system

### Supported User Types

* **AARID**: Standard business users requiring line manager approval
* **STC**: Service accounts with specific approval flows
* **Technical**: Technical service accounts with account owner approval

### Supported Role Types

* **Guardian**: Special high-privilege roles with enhanced approval
* **Monitor**: Monitoring roles with specific recertification requirements
* **High-Privilege**: Elevated access roles with additional controls
* **Standard**: Regular business roles

## Architecture

### Components

#### Core Classes

* AccessioRACFConnector.java: Main connector implementation
* GarancyAPIClient.java: SOAP client for Garancy API integration
* RACFUserManager.java: User lifecycle management
* RACFRoleManager.java: Role management and SoD validation
* ApprovalWorkflowHandler.java: Multi-level approval workflows
* RecertificationManager.java: Compliance recertification campaigns

#### SailPoint IIQ Workflows

* AccessioRACFUserCreationWorkflow.xml: User creation with approval
* AccessioRACFUserModificationWorkflow.xml: User modification workflows
* AccessioRACFUserDeletionWorkflow.xml: User deletion with cleanup
* AccessioRACFRoleAssignmentWorkflow.xml: Role assignment/removal

#### Configuration Files

* AccessioRACF.xml: SailPoint application definition
* AccessioRACFProvisioningPolicy.xml: Provisioning policies
* BeforeProvisioningRule.xml: Pre-provisioning validation
* AfterProvisioningRule.xml: Post-provisioning processing
* FieldValueRule.xml: Dynamic field population

## Installation

### Prerequisites

* SailPoint IdentityIQ 8.0 or later
* Java 8 or later
* Access to Garancy RACF system
* Maven 3.6 or later (for building)

### Build Instructions

1. **Clone the repository**:

* git clone <repository-url>  
  cd sailpoint-accessio-racf-connector

1. **Build the connector**:

* mvn clean package

1. **Deploy to SailPoint IIQ**:

* # Copy the JAR file to IIQ lib directory  
  cp target/accessio-racf-connector-1.0.0.jar $IIQ\_HOME/WEB-INF/lib/  
    
  # Copy configuration files  
  cp src/main/resources/config/\* $IIQ\_HOME/WEB-INF/config/custom/

### SailPoint IIQ Configuration

1. **Import XML configurations**:

* # Import in this order to respect dependencies  
  iiq console  
  import src/main/resources/rules/\*.xml  
  import src/main/resources/applications/AccessioRACF.xml  
  import src/main/resources/policies/AccessioRACFProvisioningPolicy.xml  
  import src/main/resources/workflows/\*.xml  
  import src/main/resources/tasks/\*.xml

1. **Configure application settings**:
   * Navigate to Applications → Accessio RACF
   * Configure Garancy endpoint, credentials, and timeout settings
   * Test the connection
2. **Schedule tasks**:
   * Configure aggregation task for regular account imports
   * Schedule recertification campaigns

## Configuration

### Environment Variables

# Garancy API Configuration  
GARANCY\_ENDPOINT=https://garancy.accessio.com/api/soap  
GARANCY\_USERNAME=sailpoint\_service  
GARANCY\_PASSWORD=encrypted\_password\_here  
GARANCY\_TIMEOUT=30000  
  
# SailPoint IIQ Configuration  
IIQ\_URL=http://localhost:8080/identityiq  
IIQ\_USERNAME=spadmin  
IIQ\_PASSWORD=admin  
  
# Connector Configuration  
CONNECTOR\_PAGE\_SIZE=100  
CONNECTOR\_BATCH\_SIZE=50  
CONNECTOR\_CACHE\_ENABLED=true  
  
# Approval Configuration  
APPROVAL\_REQUIRED=true  
APPROVAL\_MODE=serial  
APPROVAL\_TIMEOUT\_DAYS=7  
  
# SoD Configuration  
SOD\_VALIDATION\_ENABLED=true  
SOD\_CONFLICT\_ACTION=REJECT  
SOD\_HIGH\_RISK\_AUTO\_REJECT=true  
  
# Recertification Configuration  
RECERTIFICATION\_ENABLED=true  
RECERTIFICATION\_PERIODIC\_DAYS=90  
RECERTIFICATION\_GUARDIAN\_DAYS=30  
RECERTIFICATION\_MONITOR\_DAYS=30

### Application Configuration

Edit src/main/resources/applications/AccessioRACF.xml:

<Application name="Accessio RACF" type="AccessioRACF">  
 <Attributes>  
 <Map>  
 <entry key="garancyEndpoint" value="${GARANCY\_ENDPOINT}"/>  
 <entry key="garancyUsername" value="${GARANCY\_USERNAME}"/>  
 <entry key="garancyPassword" value="${GARANCY\_PASSWORD}"/>  
 <entry key="garancyTimeout" value="${GARANCY\_TIMEOUT}"/>  
 </Map>  
 </Attributes>  
</Application>

## Usage

### User Creation Workflow

1. **Initiate Request**: Submit user creation request through SailPoint
2. **Validation**: System validates user type, org unit, and role assignments
3. **SoD Check**: Automatic segregation of duties validation
4. **Approval**: Multi-level approval based on user type:
   * AARID: Line manager approval
   * STC: Org unit owner approval
   * Technical: Account owner approval
5. **Provisioning**: User created in Garancy system
6. **Notification**: Confirmation sent to requestor and approvers

### Role Assignment Workflow

1. **Request Submission**: Role assignment request initiated
2. **SoD Validation**: Check for role conflicts
3. **Special Role Handling**: Enhanced approval for Guardian/Monitor roles
4. **Approval Process**: Multi-step approval chain
5. **Role Assignment**: Role added in Garancy system
6. **Audit Logging**: Complete audit trail recorded

### Recertification Process

1. **Campaign Creation**: Automated recertification campaigns
2. **Certification Review**: Managers review user access
3. **Decision Processing**: Approve, revoke, or modify access
4. **Enforcement**: Automatic revocation of expired access
5. **Compliance Reporting**: Generate compliance reports

## API Reference

### GarancyAPIClient Methods

// Connection Management  
boolean testConnection()  
  
// User Management  
List<Map<String, Object>> getAllUsers()  
Map<String, Object> getUser(String userId)  
boolean createUser(Map<String, Object> userData)  
boolean modifyUser(String userId, Map<String, Object> changes)  
boolean deleteUser(String userId)  
boolean suspendUser(String userId)  
boolean resumeUser(String userId)  
  
// Role Management  
List<Map<String, Object>> getAllRoles()  
Map<String, Object> getRole(String roleId)  
boolean addRoleToUser(String userId, String roleId)  
boolean removeRoleFromUser(String userId, String roleId)  
List<String> getUserRoles(String userId)  
  
// Organization Unit Management  
List<Map<String, Object>> getAllOrgUnits()  
boolean orgUnitExists(String orgUnitId)

### Workflow Variables

#### User Creation Workflow

* identity: Target identity object
* userType: AARID, STC, or Technical
* orgUnit: Target organization unit
* approver: Determined approver
* workflowResult: Approval decision

#### Role Assignment Workflow

* identity: Target identity object
* requestedRoles: List of roles to assign
* currentRoles: Current user roles
* sodConflicts: List of SoD conflicts
* approvalRequired: Boolean approval flag

## Troubleshooting

### Common Issues

#### Connection Problems

Error: Unable to connect to Garancy endpoint  
Solution:   
1. Verify endpoint URL and network connectivity  
2. Check firewall settings  
3. Validate SSL certificates  
4. Review authentication credentials

#### SoD Conflicts

Error: SoD conflict detected - role assignment rejected  
Solution:  
1. Review role combination rules  
2. Request exception approval if needed  
3. Modify role assignment to resolve conflict

#### Approval Timeouts

Error: Approval request timed out  
Solution:  
1. Check approval timeout configuration  
2. Verify approver availability  
3. Review escalation settings  
4. Resend approval request if needed

### Logging Configuration

Enable detailed logging in log4j.properties:

# Connector-specific logging  
log4j.logger.com.sailpoint.connector.accessio.racf=DEBUG, connectorFile  
log4j.logger.com.sailpoint.connector.accessio.racf.GarancyAPIClient=DEBUG, apiFile  
log4j.logger.com.sailpoint.connector.accessio.racf.workflows=DEBUG, workflowFile  
  
# Audit logging  
log4j.logger.audit=INFO, auditFile

### Performance Tuning

#### Connector Performance

# Increase page size for large datasets  
connector.page.size=200  
  
# Enable caching for better performance  
connector.cache.enabled=true  
connector.cache.expiration.minutes=60  
  
# Adjust thread pool for concurrent operations  
connector.thread.pool.size=10

#### API Performance

# Increase timeout for slow responses  
garancy.timeout=60000  
  
# Configure connection pooling  
garancy.connection.pool.size=20  
garancy.connection.timeout=120000

## Security Considerations

### Authentication

* Use encrypted passwords in configuration
* Implement certificate-based authentication where possible
* Regular password rotation for service accounts

### Authorization

* Follow principle of least privilege
* Implement role-based access controls
* Regular access reviews and recertification

### Audit and Compliance

* Enable comprehensive audit logging
* Regular compliance reporting
* Secure log storage and retention

## Support

### Documentation

* API Documentation: /docs/api/
* Configuration Guide: /docs/configuration/
* Troubleshooting Guide: /docs/troubleshooting/

### Contact Information

* Technical Support: sailpoint-support@accessio.com
* Development Team: sailpoint-dev@accessio.com
* Documentation: sailpoint-docs@accessio.com

## License

Copyright (c) 2024 Accessio. All rights reserved.

This software is proprietary and confidential. Unauthorized copying, distribution, or use is strictly prohibited.

## Version History

### Version 1.0.0

* Initial release
* Full user lifecycle management
* Multi-level approval workflows
* SoD validation
* Recertification campaigns
* Comprehensive audit logging

### Planned Features

* Enhanced SoD rules engine
* Advanced reporting capabilities
* Mobile approval interface
* Integration with additional identity sources