

iMaintenance Deployment & Setup Guide

Connected Worker Solutions



Title and Copyright

Find the guide's release and publication details, copyright notices, and terms of use.

Deployment & Setup Guide for **iMaintenance**, a *Connected Office Worker Solution*.

Release Version: 2609

Release Date: 19 Sep 2026

Published Date: 19 Sep 2026

Document Version: 1.0

Copyright © 2012-2026, Innovapptive Inc. and/or its affiliates. All rights reserved.

Primary Author: Innovapptive Inc.

Copyright Notices: Neither our Application nor any content may be copied without inclusion of all copyright notices and/or disclaimers provided therein. Any third party provider logos or marks provided through the Application shall remain owned by such third party provider as may be indicated in a notice contained in the Application or content and you shall not modify or remove any such notice. Neither we nor our suppliers or any third party providers grant any rights or license to any logos, marks, or copyrighted material other than as expressly set forth herein.

PDF technology powered by PDFTron Mobile SDK copyright © PDFTron™ Systems Inc., 2001-2019, and distributed by Innovapptive Inc under license. All rights reserved.

Preface

Identify the guide's audience, text conventions, related products, and documentation contacts.

Intended Audience

This guide is intended for **SAP Basis/IT platform engineers, integration architects, Maximo administrators, and security admins** who own everything **outside the product UI**—from provisioning and data seeding to SAP or Maximo connectivity and API security.

Document Conventions

Table 0-1 Conventions followed in the document

Convention	Meaning
boldface	Indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Indicates book titles, emphasis, or placeholder variables for which you supply values.
<code>monospace</code>	Indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter

Related Products & Solutions

- [Work Order Management](#)
- [Inventory and Warehouse Management](#)
- [Analytics and Dashboards](#)

Contact Innovapptive

For information on Innovapptive products, visit the Innovapptive's Support Portal at <http://helpdesk.innovapptive.com>. The updates to this document are published on this support portal. Check this website periodically for updated documentation.

For additional information about this document, send an email to documentation@innovapptive.com.

Contents

Title and Copyright.....	ii
Preface.....	iii
1. Introduction.....	7
2. Control Network Access to iMaintenance.....	9
2.1. IP Allowlisting Through AWS WAF (Innovapptive).....	9
2.2. Allow Required Domains at the Customer Firewall.....	10
2.3. Verify and Maintain Network Access Rules.....	12
3. Connect to SAP.....	14
3.1. Install Supporting Systems and iCWP Integration Suite.....	14
3.1.1. Verify Dependent Systems Installation.....	15
3.1.2. Install and Verify iCWP Integration Suite Add-Ons.....	15
3.1.3. Activate OData Services.....	20
3.1.4. SAP Transactions and Tables Reference.....	23
3.1.5. Activate and Configure SAP NetWeaver Gateway.....	23
3.2. Configure SAP iMaintenance Roles and Authorizations.....	24
3.2.1. Use Pre-Defined SAP Roles for iMaintenance.....	25
3.2.2. Check Transport Dependencies.....	26
3.2.3. Deploy SAP iMaintenance Roles Using Transports.....	26
3.2.4. Configure SAP iMaintenance Roles.....	28
3.2.5. Define SAP iMaintenance Authorization Requirements.....	28
3.3. SAP BTP and On-Premise Integration via Cloud Connector.....	29
3.3.1. Validate Prerequisites.....	29
3.3.2. Install SAP Cloud Connector.....	29
3.3.3. Connect Cloud Connector to BTP Sub-Account.....	30
3.3.4. Configure On-Premise System Mapping.....	31
3.3.5. Add OData Service Endpoints Under Connected System.....	33
3.3.6. Verify Cloud Connector Status.....	34
3.4. Configure API Management in SAP BTP.....	34
3.4.1. Verify Prerequisites.....	35
3.4.2. Required SAP BTP API Management Roles.....	35
3.4.3. Configure API Provider.....	36
3.4.4. Create an API Proxy.....	37
3.4.5. Secure API Proxies with Policy-Based Authentication.....	38
3.4.6. Configure the Access Control Policy.....	43

3.5. Configure OAuth 2.0 Authentication.....	44
3.5.1. Create OAuth Token API Proxy.....	44
3.5.2. Verify OAuth Access Tokens.....	47
3.5.3. Remove the Verified OAuth Header Before Backend Authentication.....	48
3.5.4. Test OAuth Access to the API.....	49
3.6. Handle CSRF Tokens in API Management.....	50
4. Connect to Maximo.....	53
4.1. Check Maximo Prerequisites.....	53
4.2. Set Up Maximo Integration Components.....	54
4.2.1. Create Maximo Custom Attributes.....	54
4.2.2. Create Maximo Relationships.....	56
4.2.3. Create the Maximo Condition.....	58
4.2.4. Create the Maximo Integration User.....	58
4.2.5. Create the Integration Object Structures.....	60
4.2.6. Configure Object Structure Queries.....	97
4.2.7. Create Integration Controls.....	99
4.2.8. Create the iMaintenance Publish Channels.....	100
4.2.9. Configure Maximo Kafka Endpoints.....	132
4.2.10. Configure the Kafka Message Provider.....	134
4.2.11. Create the Maximo Queues.....	137
4.2.12. Create the Maximo Automation Scripts.....	137
4.2.13. Configure Maximo Roles and Authorizations.....	146
4.2.14. Activate Maximo Kafka Cron Tasks.....	149
4.2.15. Configure the iMaintenance External System.....	152
4.3. Connect iMaintenance to Maximo.....	155
4.4. Expose and Secure Maximo APIs.....	155
4.5. Verify the Maximo Connection.....	156
5. Initialize iMaintenance.....	157
5.1. Verify Prerequisites.....	157
5.2. Clear Existing MongoDB Collections (Optional).....	158
5.3. Seed MongoDB with Core Data.....	158
5.4. Insert SAP Data into MongoDB.....	160
5.5. Map Dependent Fields.....	161
5.6. iMaintenance Application Services.....	162
5.7. iMaintenance Scheduled Jobs.....	163
5.8. Set Up Platform Alerts and IMES for Shift Handover.....	164
5.8.1. Configure Shift Handover Kafka Consumers.....	168

6. Troubleshoot Deployment Issues..... 170

6.1. Resolve Missing OData Metadata in S/4HANA..... 170

6.2. Troubleshoot Shift Handover Alerts and IMES.....171

1. Introduction

Control network access, connect iMaintenance to SAP or Maximo, and initialize iMaintenance data and services.

This guide is written for **SAP Basis/IT platform engineers, integration architects, Maximo administrators, and security admins** who own everything **outside the product UI**, from provisioning and data seeding to SAP or Maximo connectivity and API security. Each phase below names the team that does the work. Complete the SAP or the Maximo phase, depending on the system iMaintenance connects to.

Deployment roadmap

Work through the phases in this order.

Phase	Done by	Needs	Done when
Control network access (chapter 2)	Innovapptive (AWS WAF ingress); customer network administrators (firewall egress)	Static public IP addresses or CIDRs shared with Innovapptive	Requests from an allowed IP succeed and other IPs are blocked.
Connect to SAP (chapter 3)	SAP Basis and security teams; SAP BTP administrators (Cloud Connector); SAP BTP API Management administrators	SAP Basis and Security access per the access sheet; add-on and role transport files from Innovapptive; an SAP BTP global account and sub-account with API Management and API Portal subscribed and the required API Management roles	The /INVICL/* objects appear in SE37 or SE24, the integration OData services return HTTP 200 in the SAP Gateway Client, and the iMaintenance roles appear in PFCG and are assigned in SU01; Cloud Connector shows Connected and the mapped system shows Reachable ; the API Provider Test Connection returns response code 200 or 405, the API proxy is deployed and published in a product, and the token proxy issues an OAuth access token.
Connect to Maximo (chapter 4)	Maximo administrators; Innovapptive infrastructure team (Kafka)	MAS 9 with the Integration Framework enabled; Maximo administrative access; Kafka bootstrap server, certificate, and credentials from Innovapptive	The integration objects, publish channels, endpoints, queues, scripts, and cron tasks are configured, and the iMaintenance external system is enabled.

Phase	Done by	Needs	Done when
Initialize the application and check readiness (chapters 5–6)	Innovapptive implementation team or customer IT	Active, reachable SAP or Maximo APIs; MongoDB Atlas and AWS S3 access; project repositories and seed data	Seed data is in MongoDB, the backend data load is complete (for SAP, the inbound run shows the expected status in the Integrations collection and dependent fields are mapped), and the application services and scheduled jobs are running.

What you will find in this guide

- **Control Network Access:** Enforce IP allowlisting at the infrastructure edge and allow the required domains at the customer firewall.
- **Connect to SAP:** Install the iCWP Integration Suite add-ons, activate Gateway and OData services, import roles and authorizations, connect SAP BTP through Cloud Connector, and expose and secure the APIs in SAP API Management (API keys, Access Control policy, OAuth 2.0, and CSRF tokens).
- **Connect to Maximo:** Configure MAS 9 object structures, publish channels, Kafka endpoints, queues, scripts, security, and cron tasks.
- **Initialize iMaintenance:** Check prerequisites, seed core JSONs to MongoDB, load backend data, and start application services and scheduled jobs.
- **Troubleshooting:** Fix OData metadata issues.

2. Control Network Access to iMaintenance

Coordinate Innovapptive-managed inbound IP restrictions and customer-managed outbound domain access.

Inbound traffic is restricted to trusted client IP addresses, and outbound traffic is allowed only to approved Innovapptive domains.

Responsibilities at a Glance

Layer	Responsibility
WAF (Ingress)	Managed by Innovapptive using AWS WAF to allow access only from allowlisted IPs
Firewall (Egress)	Managed by the customer to permit outbound access to specific Innovapptive domains

To maintain security and seamless communication between customer environments and Innovapptive's cloud services:

- **Ingress filtering (AWS WAF):** Only specific client IPs are allowed to access Innovapptive services.
- **Egress filtering (Customer Firewall):** Outbound traffic is restricted to the domains required for iMaintenance functionality.

This dual-layer approach prevents unauthorized access while ensuring stable, authorized connectivity across all environments.

2.1. IP Allowlisting Through AWS WAF (Innovapptive)

Configure AWS WAF rules to allow approved customer IP addresses and block other requests.

Innovapptive manages inbound access to iMaintenance services through AWS Web Application Firewall (WAF). Only customer-approved IP addresses are allowed; all others are blocked.

Prerequisites

- Enable AWS WAF on the relevant AWS resource.
- Provide a list of static public IPs or CIDRs.

1. Create an IP Set

- Navigate to the AWS WAF Console → *IP sets*.
- Click **Create IP set**.
- Enter a name (e.g., **WhitelistedIPs**).
- Select the applicable Region (Global or Regional).
- Add the customer-provided IP addresses or CIDRs.
- Click **Create**.

2. Create or Edit a Web ACL

- In the WAF Console, go to *Web ACLs*.
- Either create a new Web ACL or select an existing one.
- Attach the Web ACL to the required AWS resource (ALB, CloudFront, etc.).

3. Add an Allow Rule

- Within the Web ACL, go to *Rules* → **Add rule**.
- Rule name: **AllowOnlyWhitelistedIPs**.
- Statement: Use the previously created IP set.
- Action: **Allow**

4. Add a Block Rule

- Rule name: **BlockAllOthers**.
- Action: **Block**.
- Alternatively, set the Web ACL's default action to Block.

5. Deploy and Verify

- Ensure the Allow rule is listed **above** the Block rule in priority order.
- Save and deploy the Web ACL.
- Test access from an allowed IP (succeeds) and a non-allowed IP (blocked).

2.2. Allow Required Domains at the Customer Firewall

Allow outbound HTTPS access to the cloud domains required by your iMaintenance deployment.

The customer configures the network firewall to allow outbound access from corporate networks to these domains.

Required Domains for Outbound Access

Table 2-1 File Access (S3 Resources)

Description	Domain
-------------	--------

Table 2-1 File Access (S3 Resources) (continued)

Innovapptive file access	innoresources.s3.amazonaws.com
--------------------------	--------------------------------

Table 2-2 Authentication (Azure Active Directory)

Description	Domain
Azure AD login	login.microsoftonline.com
Legacy Azure login	login.windows.net
Microsoft auth CDN	aadcdn.msauth.net
Microsoft branding assets	aadcdn.msauthimages.net
Microsoft token service	aadcdn.msftauth.net

Table 2-3 DynamoDB Table Synchronization (AppSync)

Description	Domain
AWS AppSync endpoint	appsync-api.us-east-1.amazonaws.com
AppSync WebSocket sync	hyufofsc35b67j4grrbfmjuqv4.apps-sync-realtime-api.us-east-1.amazonaws.com

Table 2-4 MongoDB Realm Sync

Description	Domain
MongoDB Realm WebSocket	https://ws.realm.mongodb.com
MongoDB Data API	https://data.mongodb-api.com
MongoDB Realm wildcard	*.realm.mongodb.com
MongoDB API wildcard	*.mongodb-api.com

Configuration Notes

- Outbound traffic to the above domains must be allowed on TCP port 443 (HTTPS).
- Bypass SSL inspection for these domains to prevent certificate errors or service disruptions.
- Ensure DNS resolution is not restricted for these domains.
- If your firewall or proxy does not support wildcards, the wildcard entries may need to be expanded into individual domain rules.

**Note:**

This list may be updated based on service expansion or region-specific deployments.

Maximo Deployments (Kafka)

Maximo deployments also need a Kafka connection. The configuration notes above apply to the HTTPS domains, not to Kafka. Use the port and protocol supplied for the deployment.

Source	Destination	Port and protocol	Certificate and credentials
Maximo (MAS)	<bootstrap-server> supplied by Innovapptive	<port> , Kafka over TLS	Supplied by the Innovapptive infrastructure team

See [Connect iMaintenance to Maximo \(on page 155\)](#).

Summary of Responsibilities and Actions

Component	Allowlisting Type	Owner	Action Required
AWS WAF	IP-based (Ingress)	Innovapptive	Customer must share static public IPs or CIDRs
Customer Firewall	Domain-based (Egress)	Customer	Allow outbound access to specified domains on TCP 443

2.3. Verify and Maintain Network Access Rules

Test access from allowed and blocked IP addresses, review logs, and maintain network access rules.

Verification confirms that only trusted systems can reach iMaintenance services. Regular maintenance keeps the rules aligned with changing business and IT requirements.

To verify:

1. From an **allowed client IP**, call an iMaintenance API → request should succeed.
2. From a **non-allowed IP**, repeat the call → request should be blocked (HTTP 403 or connection refused).
3. Review logs for confirmation:
 - **AWS WAF / perimeter** → blocked or allowed request entries
 - **SAP Gateway / API Management** (SAP deployments) → proxy access logs
 - **Application** → unauthorized or denied attempts
 - **Maximo deployments** → the `kafkacqout` and `kafkasqout` cron task history and the queue error messages in Maximo

Maintain

1. Revalidate allowlists during major releases or quarterly reviews.
2. Update entries when client subnets, VPN egress IPs, or cloud endpoints change.
3. Remove decommissioned IPs/domains to keep rules minimal.
4. Document each entry with owner, change date, and business justification.

Next action

Avoid Common Pitfalls

- NAT or proxy may mask the client IP → ensure headers preserve the real source IP.
- Overly broad CIDRs (e.g., `/16`) expand attack surface → prefer `/32` or the smallest viable range.

3. Connect to SAP

Prepare your SAP system, connect it to SAP BTP through Cloud Connector, and expose and secure its APIs in SAP API Management.

Complete this chapter if iMaintenance connects to SAP. If it connects to Maximo, see [Connect to Maximo \(on page 53\)](#). When this chapter is complete, continue with [Initialize iMaintenance \(on page 157\)](#).

Sections

- [Install Supporting Systems and iCWP Integration Suite \(on page 14\)](#): install the iCWP Integration Suite add-ons and activate Gateway and OData services.
- [Configure SAP iMaintenance Roles and Authorizations \(on page 24\)](#): import the iMaintenance roles and set up authorizations.
- [SAP BTP and On-Premise Integration via Cloud Connector \(on page 29\)](#): connect SAP BTP to your SAP backend through Cloud Connector.
- [Configure API Management in SAP BTP \(on page 34\)](#): create the API provider and proxy, secure them, and restrict access by IP.
- [Configure OAuth 2.0 Authentication \(on page 44\)](#): issue and verify OAuth 2.0 access tokens.
- [Handle CSRF Tokens in API Management \(on page 50\)](#): fetch and pass CSRF tokens for write requests.

3.1. Install Supporting Systems and iCWP Integration Suite

Prepare SAP prerequisites, install the iCWP Integration Suite add-ons, and enable Gateway and OData services.

Related tasks

- [Verify Dependent Systems Installation \(on page 15\)](#): check the SAP components, versions, and access needed.
- [Install and Verify iCWP Integration Suite Add-Ons \(on page 15\)](#): install the add-ons for your landscape and confirm the installed objects.
- [Activate OData Services \(on page 20\)](#): activate the integration OData service, configure aliases where required, clear caches, and test the service.
- [SAP Transactions and Tables Reference \(on page 23\)](#): tables and transactions for checking data and troubleshooting.
- [Activate and Configure SAP NetWeaver Gateway \(on page 23\)](#): activate the Gateway runtime, apply the required profile parameters, and configure virus scan handling according to your organization's security policy.

When these tasks are complete, the SAP system has the iCWP add-ons installed, the Gateway framework enabled with approved security controls, and the integration OData service active.

3.1.1. Verify Dependent Systems Installation

Check the SAP components, system versions, access, and supporting services required for the integration setup.

Before beginning the installation, make sure the following prerequisites are met.

Component	Details
SAP BTP	Business Technology Platform (BTP) access
SAP Cloud Connector	Latest version installed
SAP NetWeaver Gateway	Version 7.4 or higher with SAP_GWFND 740 SP13 or above
SPAM	Version 69 or above (<i>required only for Add-On installation</i>)
SAP ECC	ECC 6.0 EHP 5 with SAP_BASIS 702 and above (recommended) Lower versions may require evaluation If the deployment option is embedded, then the NetWeaver component should be version 7.40 SAP_GWFND SP13 or above.
SAP S/4HANA	Versions 1809, 1909, 2021, 2022, 2023 Lower versions may require evaluation
Security Mechanism	Authentication compatibility requires evaluation
System Access	SAP Basis and Security access per access sheet (minimum display access)
Portal Access	Access to CWP portal with appropriate roles assigned

3.1.2. Install and Verify iCWP Integration Suite Add-Ons

Install the iCWP Integration Suite add-ons for your SAP landscape and verify the installed objects.

Use the SAP standard procedure to install the add-ons, then verify them in the ECC and Gateway systems.

Follow the steps below to install the iCWP Integration Suite add-ons in your SAP environment:

1. Obtain the latest add-ons and support packages from your Innovapptive representative.

These will typically be provided in a compressed format (e.g., .zip, .rar) via email or FTP.

2. Use the SAP standard procedure to install the add-ons.

Reference: [SAP Add-On Installation Guide](#).

3. Depending on your system landscape, refer to the appropriate table for the required components:

Table 3-1 ECC Add-On and Support Packages (Hub System)

Type	Add-On	Description	Dependency
Add-On	EWX0021310268_-0000000.PAT	iCWP integration suite 2506 ECC Add-on	None
SP01	EWX0021310268_-0000001.PAT	iCWP integration suite 2506 ECC HF01 SP	EWX0021310268_-0000000.PAT
SP02	EWX0021310268_-0000002.PAT	iCWP integration suite 2506 ECC HF02 SP	EWX0021310268_-0000001.PAT
SP03	EWX0021310268_-0000003.PAT	iCWP integration suite 2506 ECC HF03 SP	EWX0021310268_-0000002.PAT
SP04	EWX0021310268_-0000004.PAT	iCWP integration suite 2506 ECC HF04 SP	EWX0021310268_-0000003.PAT
SP05	EWX0021310268_-0000005.PAT	iCWP integration suite 2509.01 ECC SP	EWX0021310268_-0000004.PAT
SP06	EWX0021310268_-0000006.PAT	iCWP integration suite 2509.02 ECC SP	EWX0021310268_-0000005.PAT
SP07	EWX0021310268_-0000007.PAT	iCWP integration suite 2509.02.1 ECC SP	EWX0021310268_-0000006.PAT
SP08	EWX0021310268_-0000008.PAT	iCWP integration suite 2509.03 ECC SP	EWX0021310268_-0000007.PAT
SP09	EWX0021310268_-0000009.PAT	iCWP integration suite 2510.01 ECC SP	EWX0021310268_-0000008.PAT
SP10	EWX0021310268_-0000010.PAT	iCWP integration suite 2510.02 ECC SP	EWX0021310268_-0000009.PAT

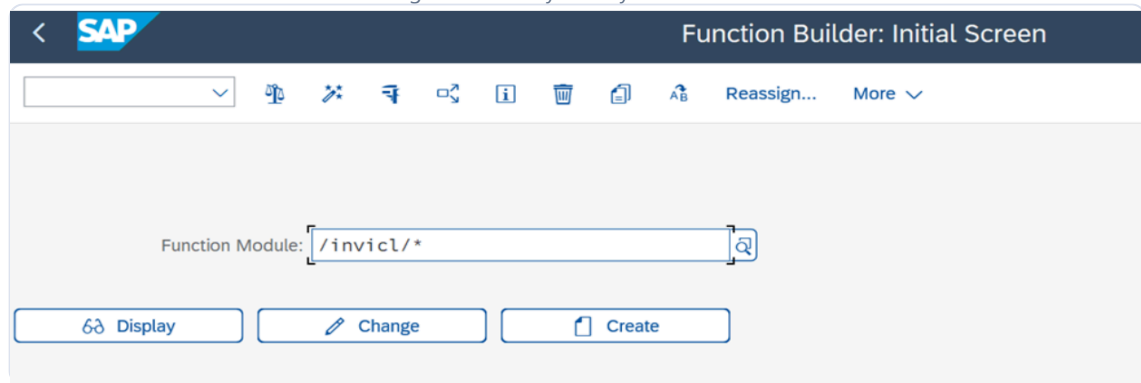
Table 3-2 Gateway Add-On and Support Packages (Hub System)

Type	Add-On	Description	Dependency
Add-On	NZY0090055495_-0000056.PAT	iCWP integration suite 2506 GW Addon	None
SP01	NZY0090055495_-0000057.PAT	iCWP integration suite 2506 HF01 GW Addon	NZY0090055495_-0000056.PAT
SP02	NZY0090055495_-0000058.PAT	iCWP integration suite 2506 HF02 GW Addon	NZY0090055495_-0000057.PAT
SP03	NZY0090055495_-0000059.PAT	iCWP integration suite 2506 HF03 GW Addon	NZY0090055495_-0000058.PAT
SP04	NZY0090055495_-0000060.PAT	iCWP integration suite 2508.4 GW Addon	NZY0090055495_-0000059.PAT

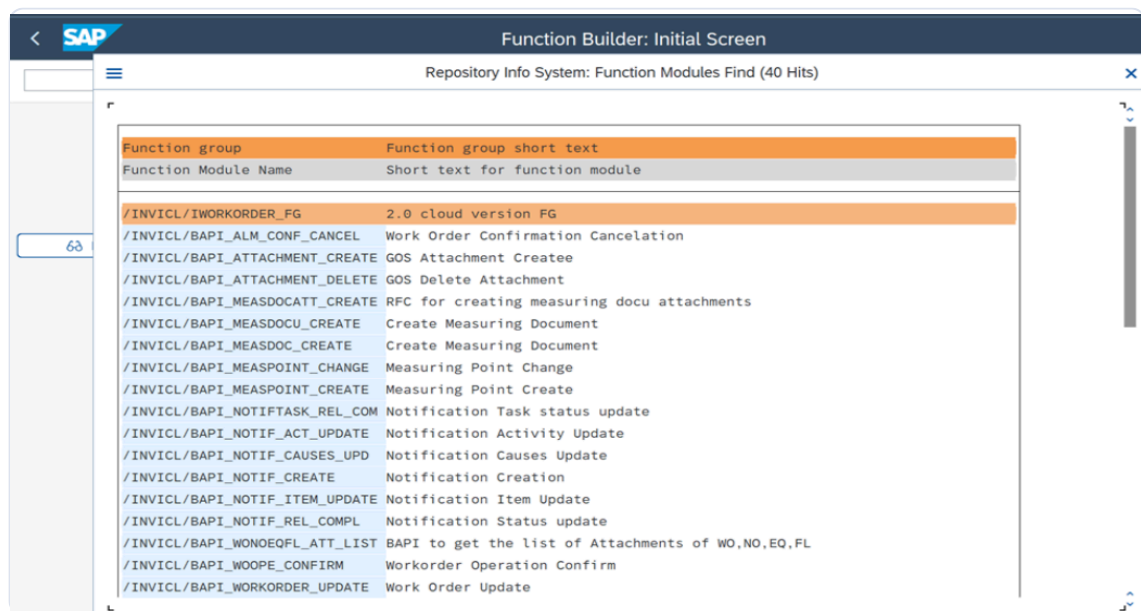
Verify Installed Add-Ons For SAP ECC Systems

1. Log in to the SAP ECC system.
2. Execute transaction code SE37.
3. In the **Function Module** field, enter /INVICL/*.

Figure 3-1 Verify ECC System Addons



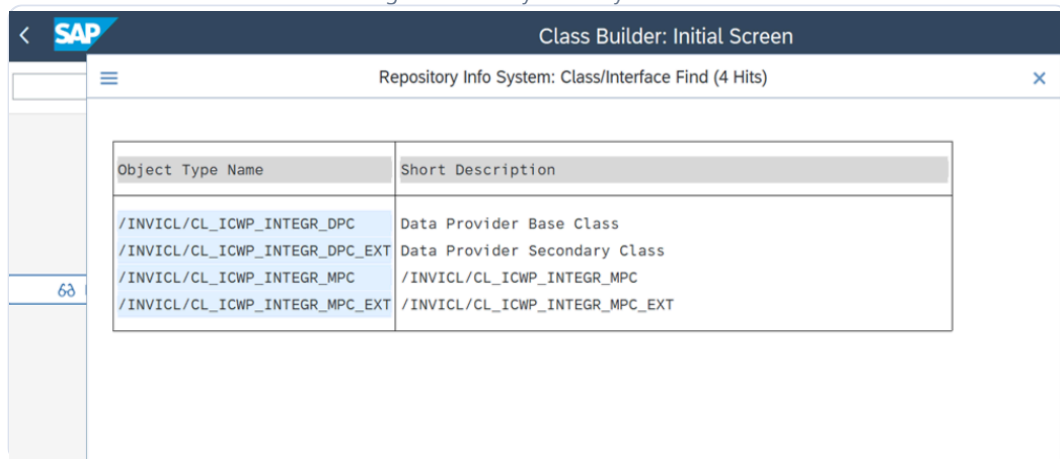
4. Press **F4** to view available function modules.
5. Capture a screenshot of the list for validation and reference.



Verify Installed Add-Ons For SAP Gateway Systems

1. Log in to the SAP Gateway system.
2. Execute transaction code SE24.
3. In the object type field, enter /INVICL/*.
4. Press **F4** to display object classes.

Figure 3-2 Verify Gateway Addons



The screenshot shows the SAP Class Builder interface. The title bar reads 'Class Builder: Initial Screen'. Below it, a search bar contains 'Repository Info System: Class/Interface Find (4 Hits)'. A table displays the search results:

Object Type Name	Short Description
/INVICL/CL_ICWP_INTEGR_DPC	Data Provider Base Class
/INVICL/CL_ICWP_INTEGR_DPC_EXT	Data Provider Secondary Class
/INVICL/CL_ICWP_INTEGR_MPC	/INVICL/CL_ICWP_INTEGR_MPC
/INVICL/CL_ICWP_INTEGR_MPC_EXT	/INVICL/CL_ICWP_INTEGR_MPC_EXT

Result

The iCWP Integration Suite add-ons are installed and confirmed in ECC and Gateway systems.

3.1.3. Activate OData Services

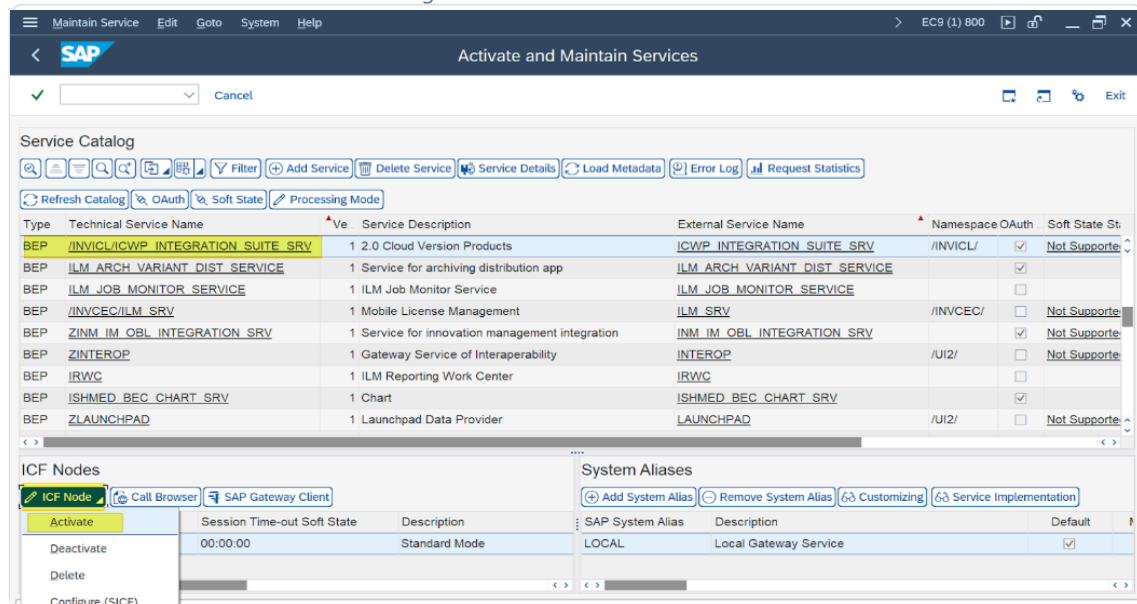
Activate the integration OData service, configure applicable aliases, clear caches, and check service responses.

After installing the iCWP add-ons, complete OData enablement in four stages to ensure endpoints are active and responding correctly.

1. Log in to the SAP Gateway system.
2. Execute transaction /IWFND/MAINT_SERVICE.
3. Locate the service /INVICL/ICWP_INTEGRATION_SUITE_SRV/.
4. From the ICF Node drop-down, select **Activate**.
5. When prompted, select the appropriate package and click **Continue**.

The package should be created with namespace Z. In some cases, a local object may be used.

Figure 3-3 Activate OData Services



Configure System Alias (If applicable)

For ECC Systems (Route-Based Implementation)

- a. Click **Add System Alias**.
- b. Choose **New Entries**.
- c. Assign the appropriate alias (e.g., <SID>) and save.



Note:

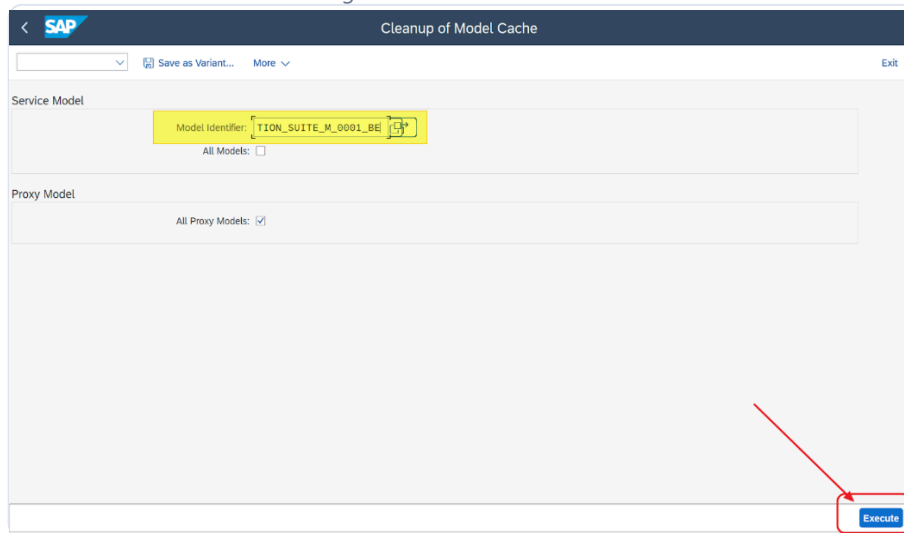
For S/4HANA Systems (Co-deployment) system alias is not required

Clean SAP Gateway Cache

1. Execute transaction `/n/iwfnf/cache_cleanup`
 - a. Select **Cleanup Cache for /INVICL/* Models** and click **Execute**.
 - b. Model Identifier: `/INVICL/ICWP_INTEGRATION_SUITE_M_0001_BE`
2. Execute transaction `/n/iwbep/cache_cleanup`.

Select **Cleanup Cache for all Models** and click **Execute**.

Figure 3-4 Clean SAP Cache



Validate OData Services

1. Execute transaction `/n/iwfnf/maint_service` in the Gateway system.
2. Filter by **Technical Service Name** using `/INVICL*`.
3. For each listed service:
 - a. Click **SAP Gateway Client**.
 - b. On the next screen, click **Execute**.
 - c. Ensure the status code **200** is returned in green.

Result

The OData service is active, required aliases (if any) are configured, caches are cleared, and the SAP Gateway Client returns HTTP 200 for each `/INVICL` service.

3.1.4. SAP Transactions and Tables Reference

Find the SAP tables and transactions used to inspect maintenance data and troubleshoot integration.

These help Basis and functional teams validate configurations, troubleshoot issues, and confirm data flows without navigating through the full SAP menu paths.

Module	Tables	Transaction Codes
Work Orders	AUFK, AFIH, AFKO, RESB, MARD, MARA	IW31, IW32, IW33, IW34, IW39, IW41, IW42, IW43, IW45, MI-GO, MMBE
Notifications	QMEL, QMIH, QMSM, QMUR, QMFE, QPCT	IW21, IW22, IW23, IW29
Equipment	EQUI, EQUZ, EQKT	IE01, IE02, IE03, IH08
Functional Locations	IFLO, IFLOT, ILOA	IL01, IL02, IL03, IH06
Measuring Points	IMPTT, IMRG	IK01, IK02, IK03, IK11, IK13, IK34
Time Sheets	CATSDB	CAT2
Usage Decisions	QALS, QAMV	QA03
Inspection Lots	QALS, QMFEL	QA13

3.1.5. Activate and Configure SAP NetWeaver Gateway

Activate SAP NetWeaver Gateway and configure its profile parameters and virus scan handling.

To activate the SAP NetWeaver Gateway:

1. In transaction SPRO, open SAP Reference IMG and navigate to SAP NetWeaver, SAP Gateway > OData Channel > Configuration > Activate or Deactivate SAP NetWeaver Gateway.
2. Select **Activate** to enable the Gateway framework.
3. Set the following profile parameters as required:

Parameter	Value / Recommendation	Purpose
login/create_sso2_ticket	2	Enable SSO ticket creation
icm/HTTPS/verify_client	1	Client certificate check. Value 1 requests a client certificate; value 2 requires one.
icm/HTTPS/verify_client_-with_issuer	CN=*	Restrict trusted issuers

Configure Virus Scan Handling

1. Open transaction **/IWFND/VIRUS_SCAN**.
2. If a virus scan profile is available, assign it to enforce file scanning for uploads and attachments.
3. If no profile is available, **Select the *Virus Scan Switched Off* check box and execute.**

Use this only as a temporary measure, with explicit approval from your IT security team.

Result

SAP NetWeaver Gateway is activated, required profile parameters are applied, and virus scan handling is configured either with a valid scan profile (recommended) or temporarily disabled under approved exception.

3.2. Configure SAP iMaintenance Roles and Authorizations

Prepare the SAP roles and authorizations used to access maintenance data and integration services.

These roles control access to master data, transactional posting, and integration endpoints.

Authorization objects ensure that only permitted users and services can perform critical actions such as creating work orders, posting goods movements, or reading plant data.

**Note:**

Missing or incomplete authorizations will result in failed API calls, restricted access to master data, or posting errors in SAP. Proper configuration at this stage ensures secure, reliable communication between iMaintenance and SAP.

Related tasks

- [Use Pre-Defined SAP Roles for iMaintenance \(on page 25\)](#)
- [Check Transport Dependencies \(on page 26\)](#)
- [Deploy SAP iMaintenance Roles Using Transports \(on page 26\)](#)
- [Configure SAP iMaintenance Roles \(on page 28\)](#)
- [Define SAP iMaintenance Authorization Requirements \(on page 28\)](#)

3.2.1. Use Pre-Defined SAP Roles for iMaintenance

Identify the supplied iMaintenance roles, transactions, and authorization objects for SAP backend and Gateway systems.

Innovapptive provides specific roles for ECC, S/4HANA, and Gateway systems. These roles include the authorizations needed for standard iMaintenance tasks.

Table 3-5 Roles for SAP Systems (ECC/S/4HANA)

Role Name	Description	Transactions	Authorization Objects
ZINV_IMAINT_ECC_END_USER	Innovapptive - iMaintenance - End User - Ecc Authorizations	CAT2, CATS_APPR_LITE, CS03, CV01N, CV02N, CV03N, IA03, IA07, IB03, IE01, IE02, IE03, IH01, IH03, IK01, IK02, IK03, IK11, IK13, IW21, IW22, IW23, IW31, IW32, IW33, IW34, IW41, IW42, IW45, MIGO	S_RFC, S_RFCACL

Table 3-6 Role for SAP Systems (S/4HANA/Gateway)

Role Name	Description	Authorizations
ZINV_IMAINT_NWG_END_USER	Innovapptive - iMaintenance - End User - Gateway Authorizations	S_RFC, S_RFCACL, S_SERVICE, S_TABU_DIS, S_USER_GRP

You can use the provided roles as is or copy them to match your internal naming conventions before generating them for use.

3.2.2. Check Transport Dependencies

Check the required transport import order and collect evidence when an import fails.

Role transports may depend on base packages. To avoid import errors, apply transports in the correct order.

- Check the transport documentation for dependency information.
- Import base transports (e.g., Innovapptive packages) before importing dependent role transports.

Troubleshoot Transport Import Issues

If issues occur during transport import, capturing logs and screenshots helps Innovapptive support resolve them quickly.

1. Take screenshots of any error messages in STMS.
2. Save the **transport import log** from the Import Queue.
3. Share the transport numbers, screenshots, and logs with your designated Innovapptive SAP Basis contact.

3.2.3. Deploy SAP iMaintenance Roles Using Transports

Import the supplied role transports, verify the roles, and add the service authorization hashes.

Importing these transports gives you a consistent role configuration across ECC, S/4HANA, or Gateway systems.

To deploy roles using transports:

1. Prepare the transport files

- Extract the archive (**.zip** or **.rar**) received from Innovapptive.
- Locate the files:
 - **Co-files** (start with K9*) → copy to **/usr/sap/trans/cofiles**
 - **Data files** (start with R9*) → copy to **/usr/sap/trans/data**

2. Import the roles using STMS

- a. Log in to the target SAP system (ECC, S/4HANA, or Gateway).
- b. Open transaction **STMS_Import**.
- c. In the Import Queue, choose **Extras > Other Requests > Add**.
- d. Enter the transport number and confirm.
- e. Select the transport request and click the **Truck** icon to start the import.
- f. Enter the **target client number**.
- g. Select the following options:
 - *Leave Transport Request in Queue for Later Import*
 - *Ignore Invalid Component Version*

- h. Confirm to begin the import process.

3. Verify the imported roles

- After import, confirm the roles are available in **PFCG**.
- In **SU01**, assign the roles to the iMaintenance technical user or other required users.

4. Update S_SERVICE authorizations

- a. Open transaction **SE16/SE16N/SE11**.
- b. Enter the table name **USOBHASH**.
- c. Use the following search parameters:

Object Type	Object Name
IWSG or IWSV	/INVICL/ICWP_INTEGRATION_SUITE_SRV_0001

- d. From the results, copy the **hashed service names** (30-character alphanumeric values).
- e. Add these values to the user's **S_SERVICE** authorization object (field: SRV_NAME).

3.2.4. Configure SAP iMaintenance Roles

Assign the supplied iMaintenance SAP roles and adapt them to the users and functions in your organization.

Innovapptive provides **pre-defined SAP iMaintenance roles** as transport requests. Import them into your SAP systems (ECC, S/4HANA, or NetWeaver Gateway) based on your deployment landscape.

Manual role creation is only required if transports are not available.

To configure SAP iMaintenance roles:

1. Import the pre-delivered transport requests into the target SAP system (ECC, S/4HANA, or Gateway) using **STMS**.
For the detailed import steps, see [Deploy SAP iMaintenance Roles Using Transports \(on page 26\)](#).
2. In transaction **SU01**, assign the imported roles to the appropriate technical or end users.
 - Adjust roles if customer-specific modifications are needed.
 - Ensure assignments align with the user's job functions.
3. If transports are not available, manually create the roles using the **transaction codes and authorization objects** listed in this guide.

3.2.5. Define SAP iMaintenance Authorization Requirements

Assign SAP roles to the required users and keep access consistent across your system landscape.

The required SAP authorizations are bundled in the Innovapptive-provided roles, and may need tailoring to your functional scope and security policies.

To assign authorizations:

1. In transaction **SU01**, assign either the **pre-delivered Innovapptive roles** or your customer-specific roles to the appropriate users.
2. Ensure access is consistent across **Development, Quality, Pre-Production, and Production** systems.
3. Review and adjust authorizations as required by your **internal security and compliance policies**.

3.3. SAP BTP and On-Premise Integration via Cloud Connector

Connect SAP BTP to your SAP backend through Cloud Connector and expose the required OData paths.

Cloud Connector lets iCWP OData services be consumed on BTP without opening inbound firewall ports.

Related tasks

1. Install and configure the SAP Cloud Connector to establish a secure tunnel between BTP and your network.
2. Map on-premise SAP systems to BTP subaccounts so ECC and S/4 systems can be registered for controlled access.
3. Define access control rules for iCWP OData services to restrict which services BTP is permitted to reach.
4. Verify that Cloud Connector is connected to the subaccount and the mapped system is reachable.

By completing these steps, you create a secure, controlled channel where iCWP services hosted on BTP can consume data from your on-premise SAP systems without compromising network security.

3.3.1. Validate Prerequisites

Check the BTP account, subaccount, and administrator access needed for Cloud Connector setup.

- A valid **SAP BTP Global Account**.
- A **subaccount** created in the Cloud Foundry environment.
- A target on-premise server for **SAP Cloud Connector**.
- Administrative credentials to access Cloud Connector.

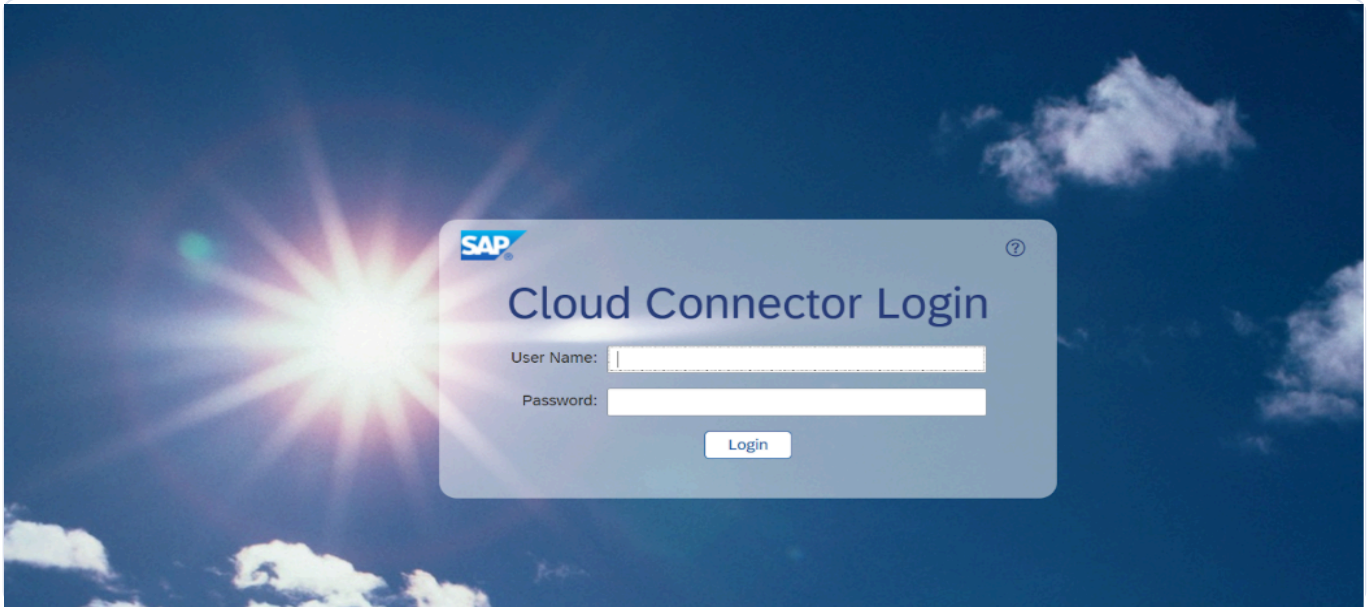
3.3.2. Install SAP Cloud Connector

Install and start SAP Cloud Connector on the designated server.

To install SAP cloud connector:

1. Get the latest SAP Cloud Connector version from the **SAP Software Download Center**.
2. Launch the installer and follow the on-screen instructions to complete the installation.
3. Start the Cloud Connector service.
4. Open the Cloud Connector UI in a browser by entering the host address and port number, for example: `https://<Host Address>:<Port>`
5. Log in using the default administrator credentials.

Figure 3-5 Log in to SAP Cloud Connector



3.3.3. Connect Cloud Connector to BTP Sub-Account

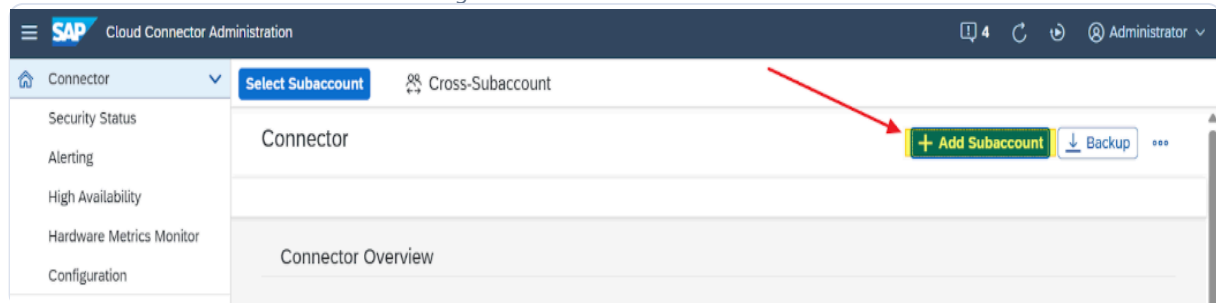
Register Cloud Connector with the SAP BTP subaccount used for the integration.

SAP Cloud Connector must be installed and running. See [Install SAP Cloud Connector \(on page 29\)](#).

To connect cloud connector to BTP sub-account:

1. In the Cloud Connector UI, click **Add Sub-account**.

Figure 3-6 Connect to Sub-Account

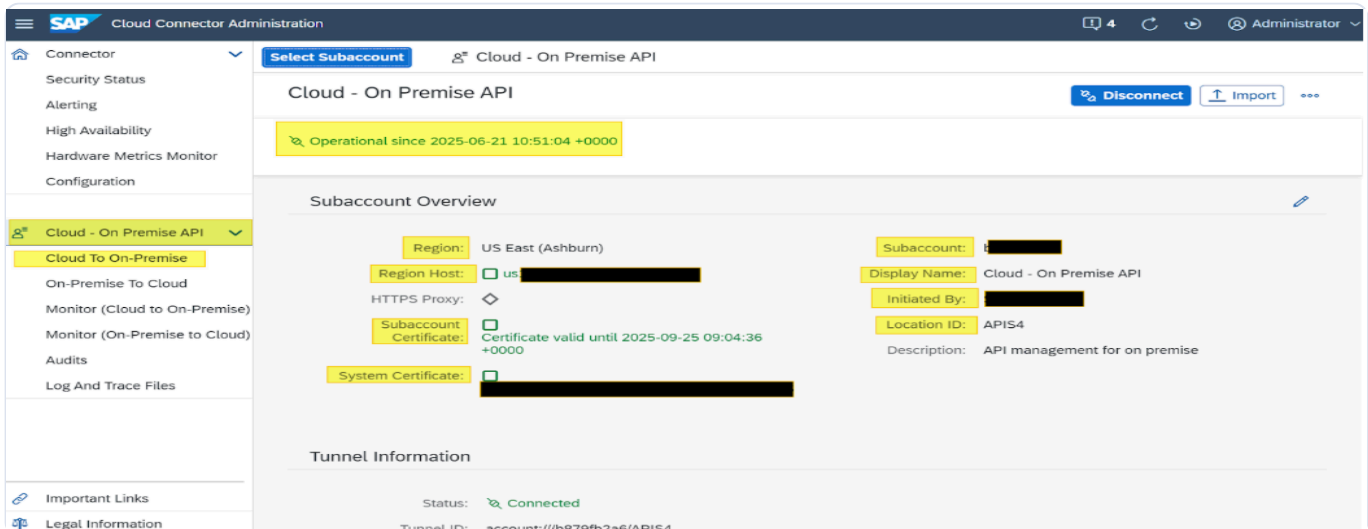


2. Enter the following details:

- **Region:** Select the region matching your BTP subaccount (e.g., eu10, us10)
- **Subaccount ID:** Available in the BTP cockpit under *Subaccount Overview*
- **Subaccount Name:** (Optional, for internal reference)
- **Subaccount User:** A user with at least **Subaccount Administrator** role
- **Password:** Password for the subaccount user
- **Location ID** (*optional*): Use if multiple Cloud Connectors are configured for redundancy or load balancing.

3. Click **Save** to establish the connection.

You will see the screen showing the **Subaccount** details and confirmation that the connection to SAP BTP has been established.



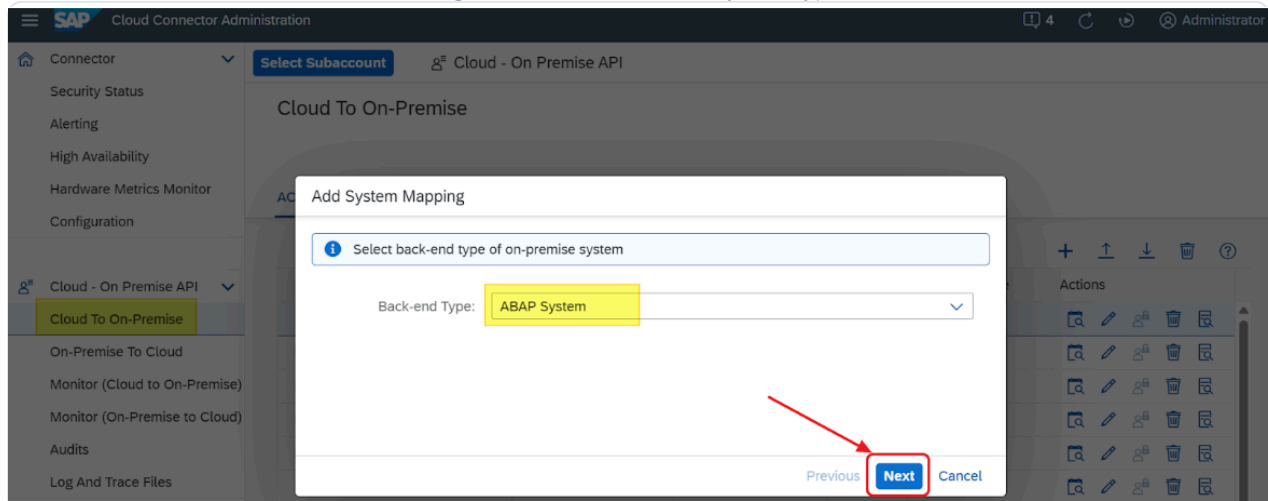
3.3.4. Configure On-Premise System Mapping

Map the internal SAP host and port to the virtual address used by SAP BTP.

To configure on-premise system mapping:

1. Open the Cloud Connector UI in a browser.
2. In the **Subaccount** view, select **Cloud To On-Premise**
3. Click **Add System Mapping**.
4. Choose the backend system type:
 - Select **ABAP System** (for SAP Gateway/OData).
 - or **SAP HANA** if exposing HANA endpoints.

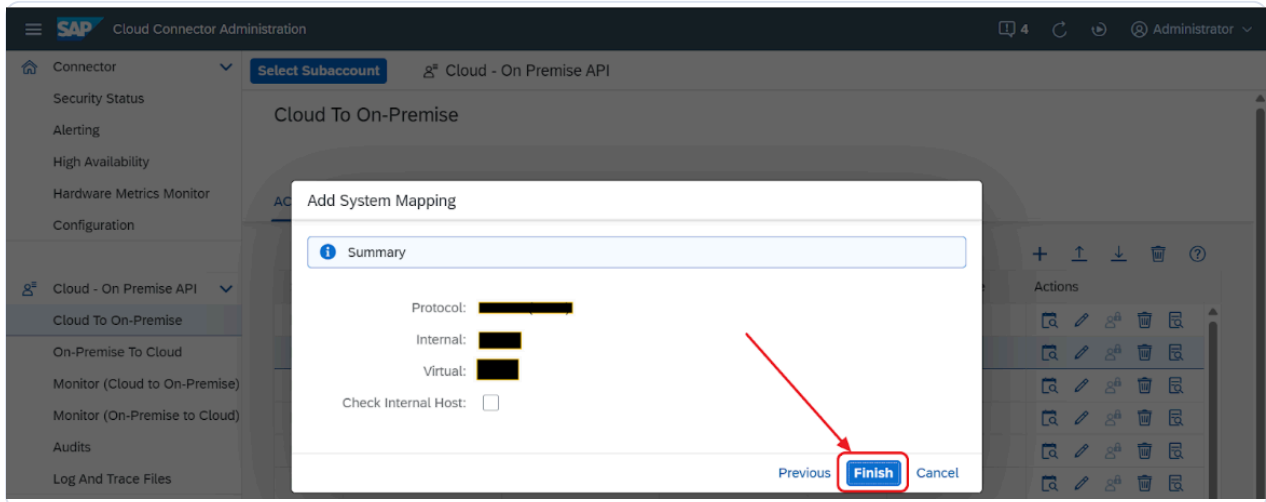
Figure 3-7 Choose Backend System Type



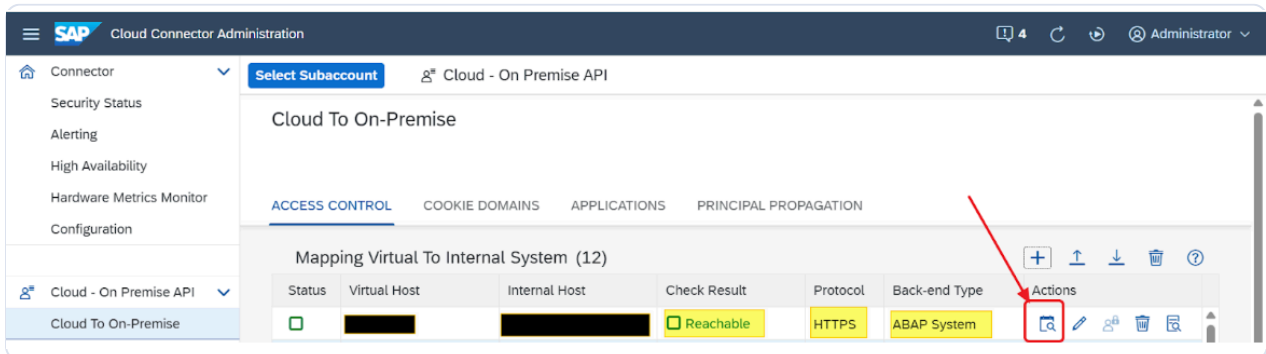
5. Select **HTTPS** as the protocol.
6. Enter the **internal host/IP address and port** of the target **RISE with SAP** system hosted on the hyperscaler.
7. Define the **Virtual Host** and **Virtual Port**. For example: `<Virtual Hostname> <Port>`.
8. Click **Next**.
9. Leave **Principal Propagation** and **System Certificate for Logon** disabled unless explicitly required.
10. In the **Host in Request Header** field, select **Use Virtual Host**.
11. Enter **Description** (optional).

12. Review the configuration and click **Finish**.

13. If the configuration is correct, the status shows **Reachable**.



14. Click the **search icon** under the **Actions** column to check if the system is reachable.



3.3.5. Add OData Service Endpoints Under Connected System

Allow SAP BTP to reach the required OData paths through the mapped Cloud Connector system.

To add OData service endpoints under connected system:

1. Navigate to the **Resources of <External Virtual Host>** tab under your mapped system.
2. Click **+ Add Resource**.
3. Enter the required details: **URL Path**: For example, `/sap/opu/odata/inviol/`.
4. Select Path And All Sub-Paths.
5. Click on Save the access control settings.

**Note:**

This step ensures that only specific paths and services are exposed to SAP BTP.

3.3.6. Verify Cloud Connector Status

Check that Cloud Connector is connected to the BTP subaccount and the mapped system is reachable.

To test the connection from BTP:

1. Log in to the SAP BTP Cockpit.
2. Navigate to your subaccount → **Connectivity** → **Cloud Connectors**.
3. Confirm that:
 - The Cloud Connector status shows **Connected**
 - The mapped system is visible and marked **Reachable**

3.4. Configure API Management in SAP BTP

Set up API providers and proxies, secure consumer access, and publish APIs for registered applications.

Related tasks

- [Verify Prerequisites \(on page 35\)](#)
- [Required SAP BTP API Management Roles \(on page 35\)](#)
- [Configure API Provider \(on page 36\)](#)
- [Create an API Proxy \(on page 37\)](#)
- [Secure API Proxies with Policy-Based Authentication \(on page 38\)](#)
- [Configure the Access Control Policy \(on page 43\)](#)

3.4.1. Verify Prerequisites

Check the BTP subaccount, Cloud Connector connection, and API Management services needed before creating proxies.

- A subaccount created in **SAP BTP (Cloud Foundry environment)**
- **SAP Cloud Connector** already set up and connected to your subaccount
- The following services subscribed under your subaccount:
 - **API Management, API Portal**
 - **Cloud Connector** (if exposing on-premise APIs)

3.4.2. Required SAP BTP API Management Roles

Identify the BTP and API Management roles used to administer providers, proxies, products, and applications.

These roles control who can configure API providers, create proxies, and manage policies for iCWP services.

Without these roles, key tasks such as creating API providers or deploying proxies cannot be performed.

Role Name	Purpose
Subaccount Viewer	Read-only access to the subaccount
Subaccount Service Administrator	Administrative access to service brokers and environments on a subaccount level
APIManagement.SelfService.Administrator	API Portal Onboarding
APIPortal.Administrator	APIPortalAdmin
APIPortal.Developer	Manages API proxies and products with read-only access to all other entities.
APIPortal.Configurator	Manages entities such as API providers, certificates, and rate plans.
APIPortal.Service.CatalogIntegration	CatalogIntegration
AuthGroup.SelfService.Admin	Self Service Admin developer hub
AuthGroup.API.Admin	Manages Users and applications in SAP API Management developer hub
AuthGroup.ContentAutho	Manages Content

AuthGroup.API.ApplicationDeveloper	Application Developer in SAP API Management developer hub
AuthGroup.Content.Admin	Manages Content in SAP API management developer hub
AuthGroup.Site.Admin	Manages Site updates in SAP API management developer hub
AuthGroup.APIPortalRegistration	Role to create API Portal Connection Request

*Optional: CatalogIntegration is required only for Business Hub publishing.

3.4.3. Configure API Provider

Define the Cloud Connector connection and catalog settings used by API Management to discover SAP OData services.

The provider stores the virtual host, port, and connectivity settings that proxies use to reach iCWP OData services through Cloud Connector.

To configure API provider:

1. Navigate to your BTP subaccount in the cockpit.
2. Navigate to **Instances and Subscriptions**.
3. Under **API Management, API Portal**, click **Go to Application**.
4. Navigate to **Configure, API Providers** in the API Portal dashboard.
5. Click **Create** and fill in the following details:
 - **Name**: Any identifier for your API provider.
 - **Description**: Provide description (This is optional).
6. Navigate to the **Connection** tab and enter the following details:
 - **Type**: Select **On Premise**
 - Enter the **Virtual Host, Port, and Location ID** that were configured during the Cloud Connector setup, where the BTP subaccount and the on-premise system were added. Refer to the topic [Configure On-Premise System Mapping \(on page 31\)](#)
 - **Authentication** and **Additional Properties** should be **None**

7. Navigate to the **Catalog Service Settings** and enter the following:

- **Path Prefix:** /sap/opu/odata
- **Service Collection URL: For ECC - /iwfnd/catalogservice/ServiceCollection For S/4 HANA - /IWFND/CATALOGSERVICE;v=2/ServiceCollection**
- **Authentication Type:** Basic
- **Username:** System UserID
- **Password:** System UserID Password

8. Click **Save**.

9. After saving the configuration successfully, follow these steps:

a. Click **Test Connection**.

- If the connection is successful, one of the following messages is displayed:
 - **Connection to the system was successful with response code: 200; Message: OK.**
 - **System is up and reachable. However, the ping check responded with code: 405; Message: Method Not Allowed.**
- Both messages indicate a successful connection.
- If neither of these messages appears, it means the configuration details are incorrect.

3.4.4. Create an API Proxy

Create and deploy an API proxy for the iCWP Integration Suite OData service.

The proxy uses the API Provider details and makes the service available for policy enforcement and external consumption.

Create an API Proxy to act as a secure, managed layer between external consumers and your backend service.

1. In the API Portal, navigate to **Develop > APIs**.

2. Click **Create**.

3. In the **Create API** dialog:

- a. Select **API Provider** as the source.
- b. Select the **API Provider** created in the previous step.
- c. Click the **Discover** button.
- d. Choose **OData** service name and click OK (OData: **ICWP_INTEGRATION_SUITE_SRV**).

4. Click **OK** and provide **API Name**, **Title** and **Version**.

5. Click **Create**.

6. Click **Save**.

7. Click **Deploy**.
8. Select the API proxy, click on the **three dots in the top-right corner**, and choose **Synchronize**.
9. Click **Yes** and then **Deploy**.

3.4.5. Secure API Proxies with Policy-Based Authentication

Authenticate external API consumers with an API key or OAuth token and use stored service credentials for backend access.

If no policies are configured, users are prompted for SAP credentials by default.

SAP BTP API Management supports two authentication methods for external consumers:

- **API Key**
- **OAuth 2.0**

These ensure secure access between Innovapptive Cloud and SAP BTP, while communication between **SAP BTP API Management** and the **SAP backend system (SAP Gateway)** continues to use Basic Authentication with a dedicated service **Service User ID** (system user).

To enable API Key-based authentication in SAP BTP API Management, follow the steps below:

1. Create a Key Value Map.
2. Define API Key validation policies.
3. Assign API proxies to a product and publish it.
4. Create an application and subscribe it to the product.
5. Share the API key with the external consumer.
6. Test the API in Postman using the API key.

3.4.5.1. Create a Key Value Map

Store the SAP service user ID and password as encrypted values for use by backend authentication policies.

The values are encrypted and later referenced in authentication policies.

To create a key value map:

1. In the API Management console, click **Configure** from the left panel.
2. Click the **Key Value Maps** tab and click **Create**.
Name the key value map **iCWPcredentials**. The Key Value Map Operations policy in [Define API Key Validation Policies \(on page 39\)](#) reads the credentials from this name (**mapIdentifier**).
3. Click **Add** and enter these keys:
 - **UserID** <SAP Service User ID>
 - **Password** <SAP Service User Password>
4. Select **Encrypt Key Value** for both entries.
5. Click **Save**.

3.4.5.2. Define API Key Validation Policies

Validate the consumer's API key, retrieve stored SAP credentials, and set backend Basic authentication.

Only clients with a valid key can access your iCWP services.

Define and Verify Key Policy

Add a Verify API Key policy to your API Proxy. This policy checks the API key passed in the **x-api-key** header against the keys associated with registered applications in the Developer Portal. Only authorized consumers with valid keys can access the API. The policy is typically applied in the **PreFlow** of the **ProxyEndpoint**.

To define and verify key policy:

1. In the API Management console, click **Develop**.
2. Select the relevant **API Proxy**.
3. Open the **Policies** tab and click **Edit**.
4. In the **ProxyEndpoint**, select **PreFlow**.
5. From **Security Policies**, select **Verify API Key** and click **+** to add it.
6. Enter the **Policy Name**, set **Stream = Incoming Request**, and click **Add**.
7. In the policy editor, replace the default content with the following XML

Policy Message:

```
<!-- Specify in the APIKey element where to look for the variable containing the api key -->
<VerifyAPIKey async='true' continueOnError='false' enabled='true'
  xmlns='http://www.sap.com/apimgmt'>
  <APIKey ref='request.header.x-api-key'/>
</VerifyAPIKey>
```

8. Click **Update**.

9. Click **Save**, then **Redeploy** the API Proxy so the policy takes effect.

Define the Key Value Map Operations Policy

Add a Key Value Map (KVM) Operations policy to your API Proxy. This policy retrieves stored credentials (for example, the SAP service user ID and password) from the Key Value Map and makes them available for downstream policies.

1. In the API Management console, select the relevant **API Proxy**.
2. Open the **Policies** tab and click **Edit**.
3. In the **ProxyEndpoint**, select **PreFlow**.
4. From **Mediation Policies**, select **Key Value Map Operations** and click **+**.
5. Enter the **Policy Name**, set **Stream = Incoming Request**, and click **Add**.
6. In the policy editor, replace the default content with the following XML:**Policy Message:**

```
<!-- Key/value pairs can be stored, retrieved, and deleted from named existing maps by configuring this
policy by specifying PUT, GET, or DELETE operations -->
<!-- mapIdentifier refers to the name of the key value map -->
<KeyValueMapOperations mapIdentifier="iCWPcredentials" async="true" continueOnError="false"
  enabled="true" xmlns="http://www.sap.com/apimgmt">
  <!-- PUT stores the key value pair mentioned inside the element -->
  <Get assignTo="private.username">
    <Key>
      <Parameter>UserID</Parameter>
    </Key>
  </Get>
  <Get assignTo="private.password">
    <Key>
      <Parameter>Password</Parameter>
    </Key>
  </Get>
  <!-- the scope of the key value map. Valid values are environment, organization, apiproxy and policy
  -->
  <Scope>environment</Scope>
</KeyValueMapOperations>
```

7. Click **Update**.

8. Click **Save**, then **Redeploy** the API Proxy so the policy is active.

Define the Basic Authentication Policy

Add a Basic Authentication policy to your API Proxy. This policy encodes the SAP service user credentials (retrieved from the Key Value Map) into the **Authorization** header, enabling secure downstream calls from API Management to the SAP backend.

1. In the API Management console, select the relevant **API Proxy**.
2. Open the **Policies** tab and click **Edit**.
3. In the **ProxyEndpoint**, select **PreFlow**.
4. From **Security Policies**, select **Basic Authentication** and click **+**.
5. Enter the **Policy Name**, set **Stream = Incoming Request**, and click **Add**.
6. In the policy editor, replace the default content with the following XML:

Policy Message

```
BasicAuthentication async='true' continueOnError='false' enabled='true'  
<!-- Operation can be Encode or Decode -->  
<Operation>Encode</Operation>  
<IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>  
<!-- for Encode, User element can be used to dynamically populate the user value -->  
<User ref='private.username' />  
<!-- for Encode, Password element can be used to dynamically populate the password value -->  
<Password ref='private.password' />  
<!-- Source is used to retrieve the encoded value of username and password. This should not be used if  
the operation is Encode-->  
<Source>request.header.Authorization</Source>  
<!-- Assign to is used to assign the encoded value of username and password to a variable. This should  
not be used if the operation is Decode -->  
<AssignTo createNew='false'>request.header.Authorization</AssignTo>  
</BasicAuthentication>
```

7. Click **Update**.
8. Click **Save**, then **Redeploy** the API Proxy to apply the changes.

3.4.5.3. Create a Product and Bundle the API

Bundle the integration API in a product and publish it for application subscriptions.

The product makes the iCWP API available in Developer Hub.

To create a product and bundle the API:

1. Open the **API Portal** and select **Develop** on the left-hand panel.
2. Click on the **Products** tab and click **Create** to start creating a new product.
3. Enter the Product Name as **<iCWPIntegrationSuite>** and the Title as **<iCWP Integration Suite>**.

4. Click the **APIs** tab and click **Add** to view the list of deployed APIs.
5. Select the API **ICWP_INTEGRATION_SUITE_SRV** and click **OK**.
6. Click **Publish** to make the product available in the **Developer Hub (API Business Hub Enterprise)**.
7. Click on More icon on top-right corner.

You will see two options: **Client SDK** and **Developer Hub**. Click on **Developer Hub** to view the published products.

The **Developer Hub portal** opens with the **Published Products** list.

3.4.5.4. Create a Developer Hub Application

Create a Developer Hub application and subscribe it to the published integration product.

To create an application to generate application key and secret key:

1. In Developer Hub, select the **published product**.
2. Click the **Subscribe** drop-down and select **Create New Subscription for Application**.
3. Enter **Application Name** = `<iCWPIntegrationSuite>` and **Short Text** = `<iCWP Integration Suite>`

3.4.5.5. Test API Proxy with API Key Authentication

Test API proxy access with the application key supplied by Developer Hub.

To test API proxy with API key authentication:

1. Open **API Portal**.
2. Click **Develop** on left-side panel and copy or note the **API Proxy URL** on the right-side.
3. Click the More icon on top-right and select **Developer Hub**.
4. Copy or note the **Application Key**.

5. Open **Postman**.
6. In Postman, set **Auth Type** to **API Key**.
 - Key: **x-api-key**
 - Value: **<Application Key>** (copied from the Developer Hub)
7. Set the **Request URL** to the **API Proxy URL** (copied from the API Portal).

Case 1: Passing correct API key to the Authentication

Case: 2 Passing Incorrect API key to the Authentication

3.4.6. Configure the Access Control Policy

Apply IP access restrictions to an API proxy with SAP API Management's Access Control policy.

Before you begin

When to use

- You need proxy-level control/auditing per API.
- Compliance requires layered enforcement beyond WAF/firewall.
- Temporary, API-specific allowlists are needed without changing perimeter rules.

This adds IP allowlisting at the API proxy layer, as an additional control behind the WAF and firewall.

This task needs an existing API proxy. See [Create an API Proxy \(on page 37\)](#).

Follow the steps below to apply the AccessControl policy in your API Proxy:

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Traffic Management Policies**, choose **Access Control**.
6. Click on the + button to add the policy.
7. Enter the **Policy Name**, set the Stream as Incoming Request and click **Add**.

8. Click **Save**.
9. Deploy this revision
10. Test from allowed vs. non-allowed IPs to confirm behavior.

Next action

Important Considerations

- Enforcement is **per-proxy**; perimeter/WAF rules still apply first.
- Use `/32` for exact IPs; use CIDRs (e.g., `/24`) only when justified.
- Ensure Cloud Connector/API Gateway preserves the client IP (e.g., `X-Forwarded-For`) so the policy matches the real source.
- Keep proxy allowlists in sync with perimeter rules to avoid drift.

3.5. Configure OAuth 2.0 Authentication

Configure token generation and verification so external consumers can call protected integration APIs.

To enable OAuth authentication for an API Proxy, configure the following activities:

- **Create an OAuth Token API Proxy** – define the token endpoint using an OAuth v2.0 policy in *GenerateAccessToken* mode.
- **Create the actual API Proxy** – secure it with a *VerifyOAuthToken* policy and configure backend authentication.
- **Publish the API as a Product and create an Application** – generate the Client ID and Client Secret needed for token requests.
- **Test the OAuth flow** – generate a token and invoke the iCWP API with it.

3.5.1. Create OAuth Token API Proxy

Set up the token proxy, apply token generation, publish it in the API product, and test token issuance.

The token proxy issues the access tokens that external consumers use to call protected iCWP APIs.

- Create an API Proxy in API Management (use a dummy URL as backend).
- Add OAuth v2.0 Policy in `GenerateAccessToken` mode.
- Deploy the proxy – this acts as the token endpoint.
- Add OAuth Token Generation API Proxy to the Product and Republishing
- Test OAuth Token Generation using the API proxy.

3.5.1.1. Create the OAuth Token Proxy

Create the API proxy used as the OAuth token endpoint and configure it without a target route.

The proxy uses a dummy backend and serves as the OAuth token endpoint for iCWP APIs.

To create an API proxy in API management:

1. In the **API Portal**, navigate to **Develop > APIs**.
2. Click **Create**.
3. In the **Create API** dialog:
 - a. Select **URL** as the source.
 - b. Enter Working Dummy URL: < <https://www.innovapptive.com/> > (it can be any working URL).
 - c. Enter **Name** : **ICWPGenerateoAuthToken** and **Title**: **ICWP Integration Suite Generate oAuth Token** for the **API Proxy**.
 - d. Provide **API Base Path**:**/token**.
 - e. Click **Create** to generate the proxy.
4. Go to **Proxy Endpoint** → **Route Rules**:
 - a. Route Rule Name: **Default**.
 - b. Target Endpoint: **None**.
 - c. Click **Save**.

3.5.1.2. Configure OAuth Access Token Generation

Generate OAuth access tokens with the client credentials grant.

The OAuth v2.0 policy in `GenerateAccessToken` mode issues access tokens with the `client_credentials` grant type and returns them in the response.

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Security Policies**, choose **OAuth v2.0**.
6. Click on the **+** button to add the policy.
7. Enter the **Policy Name**, set the **Stream** as **Incoming Request** and click **Add**.
8. Replace the existing default content in the Body section with the following policy XML and click **Update**.

Policy Message:

```
<OAuthV2 async="false" continueOnError="false" enabled="true" xmlns="http://www.sap.com/apimgmt">
  <!-- By default, VerifyAccessToken expects the access token to be sent in an Authorization header. You
  can change that default using this element<AccessToken> -->
  <!-- if you want to pass access token in an customer header "access_token": -->
  <!-- <AccessToken>request.header.access_token</AccessToken> -->
  <!-- If you want to pass access token in query param "access_token": -->
  <!-- <AccessToken>request.queryparam.access_token</AccessToken> -->
  <!-- this flag has to be set when you want to work with third-party access tokens -->
  <ExternalAuthorization>false</ExternalAuthorization>
  <!-- valid values are GenerateAccessToken, GenerateAccessTokenImplicitGrant, GenerateAuthorizationCode ,
  RefreshAccessToken , VerifyAccessToken , InvalidateToken , ValidateToken -->
  <Operation>GenerateAccessToken</Operation>
  <GenerateResponse enabled="true"/><SupportedGrantTypes>
    <GrantType>client_credentials</GrantType>
  </SupportedGrantTypes>
  <Tokens/>
</OAuthV2>
```

9. Click **Save**.
10. In the **API Portal**, navigate to **Develop**.
11. Click **Click to Deploy** to apply the changes.

3.5.1.3. Add OAuth Token Proxy to Product and Republish

Add the token-generation proxy to the integration product and republish it.

After republishing, both the API proxy and the token proxy are available in Developer Hub.

To add OAuth token generation API proxy to product:

1. In the **API Portal**, navigate to **Develop** → **Products** tab,
2. Click on the already created product **iCWPIntegrationSuite**.
3. Go to the **APIs** tab and click **Edit**.
4. Click **Add**.
5. In the API list, select the OAuth token generation API proxy: **ICWPGenerateoAuthToken** and click **OK**.
6. Click **Publish**.

3.5.1.4. Test OAuth Token Generation using API Proxy

Request an OAuth access token in Postman using the Developer Hub client ID and secret.

This confirms that the token proxy issues access tokens for the client ID and secret from Developer Hub.

To test OAuth token generation using an API proxy:

1. Open **Postman** and go to the **Authorization** tab.
2. Set **Auth Type** to **OAuth 2.0**.
3. Fill in the following details:
 - a. **Token Name**: any descriptive name
 - b. **Grant Type**: *Client Credentials*
 - c. **Access Token URL**: URL of the **Generate OAuth Token API Proxy**
 - d. **Client ID**: Application Key from **Developer Hub**
 - e. **Client Secret**: Secret Key from **Developer Hub**
4. Scroll down and click **Get New Access Token**.

3.5.2. Verify OAuth Access Tokens

Verify the incoming OAuth access token before allowing access to the protected API.

Only requests with a valid token issued by the token proxy can access iCWP APIs.

To add OAuth v2.0 policy:

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Security Policies**, choose **OAuth v2.0**.
6. Click on the **+** button to add the policy.
7. Enter the **Policy Name**, set the **Stream** as **Incoming Request** and click **Add**.
8. Replace the existing default content in the Body section with the following policy XML and click **Update**.

Policy Message:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAuthV2 async="false" continueOnError="false" enabled="true" xmlns="http://www.sap.com/apimgmt">
  <!-- By default, VerifyAccessToken expects the access token to be sent in an Authorization header. You
  can change that default using this element<AccessToken> -->
  <!-- if you want to pass access token in an customer header "access_token": -->
  <!-- <AccessToken>request.header.access_token</AccessToken> -->
  <!-- If you want to pass access token in query param "access_token": -->
  <!-- <AccessToken>request.queryparam.access_token</AccessToken> -->
  <!-- this flag has to be set when you want to work with third-party access tokens -->
  <ExternalAuthorization>false</ExternalAuthorization>
  <!-- valid values are GenerateAccessToken, GenerateAccessTokenImplicitGrant, GenerateAuthorizationCode ,
  RefreshAccessToken , VerifyAccessToken , InvalidateToken , ValidateToken -->
  <Operation>VerifyAccessToken</Operation>
  <GenerateResponse enabled="true"/><SupportedGrantTypes/>
  <Tokens/>
</OAuthV2>
```

9. Click **Save**.

The **Verify OAuth Access Token** policy should be placed **first in the execution flow**. You can change its position by clicking on the **navigation arrows**.

3.5.3. Remove the Verified OAuth Header Before Backend Authentication

Remove the verified OAuth Authorization header before the request uses backend Basic authentication.

OAuth 2.0 secures external access at the SAP BTP API Management layer, while API Management still calls the SAP backend with Basic Authentication from the Key Value Map. This policy removes the OAuth token from the request after verification, so the request is forwarded cleanly to the backend.

To add policy to remove OAuth token:

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Mediation Policies**, choose **Assign Message**.
6. Click on the **+** button to add the policy.
7. Enter the **Policy Name**, set the **Stream** as **Incoming Request** and click **Add**.
8. Replace the existing default content in the Body section with the following policy XML, and click **Update**.

Policy Message:

```
<!-- This policy can be used to create or modify the standard HTTP request and response messages -->
<AssignMessage async="false" continueOnError="true" enabled="true" xmlns='http://www.sap.com/apimgmt'>
  <Remove>
    </Headers>
  </Remove>
  <Header name="Authorization"></Header>
  </Headers>
  </Remove>
  <IgnoreUnresolvedVariables>true</IgnoreUnresolvedVariables>
  <AssignTo createNew="false" type="request"></AssignTo>
</AssignMessage>
```

9. Click **Save**.

3.5.4. Test OAuth Access to the API

Call the protected API with a Bearer token and check valid, invalid, and expired token responses.

To test OAuth authentication using Postman:

1. Open Postman and set the Authorization Type to **Bearer Token**.
2. Provide the OAuth token (you can obtain this token from the previous steps explained in the document).
3. If the token is valid, you will receive a successful response with status code **200 OK**.
4. If the token is invalid, the response code will be **401 Unauthorized**.
5. If the token is expired, the response code will be **401 Unauthorized**.

3.6. Handle CSRF Tokens in API Management

Fetch a backend CSRF token and pass it with the response cookies on applicable write requests.

For **POST**, **PUT**, and **DELETE** requests, configure a ServiceCallout **scCSRF** policy to automatically fetch and pass the required X-CSRF token from the backend.

To handle CSRF token in API management:

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Extension Policies**, choose **Service Callout**.
6. Click on the + button to add the policy.
7. Enter the **Policy Name**, set the Stream as Incoming Request and click **Add**.

8. Replace the existing default content in the Body section with the following policy XML and click **Update**.

Condition String:

Policy Message:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<ServiceCallout async="true" continueOnError="false" enabled="true" xmlns="http://www.sap.com/apimgmt">
  <Request>
    <Set>
      <Headers>
        <Header name="x-csrf-token">fetch</Header>
        <Header name="Authorization">{request.header.Authorization}</Header>
      </Headers>
      <Verb>GET</Verb>
    </Set>
    <IgnoreUnresolvedVariables>false</IgnoreUnresolvedVariables>
  </Request>
  <Response>callOutResponse</Response>
  <Timeout>30000</Timeout>
</HTTPTargetConnection>
  <APIProvider>EW8API</APIProvider>
  <Path>/sap/opu/odata/invc1/ICMP_INTEGRATION_SUITE_SRV</Path>
</HTTPTargetConnection>
</ServiceCallout>
```



Note:

EW8API is an example API Provider name. Replace it with the API Provider you created in [Configure API Provider \(on page 36\)](#), for example:

```
<HTTPTargetConnection>
  <APIProvider>[Provide API Provider Name here]</APIProvider>
  <Path>/sap/opu/odata/invc1/ICMP_INTEGRATION_SUITE_SRV</Path>
</HTTPTargetConnection>
```

9. Click **Save**.
10. Under **Mediation Policies**, choose **Assign Message** and click the + button.

11. Enter the **Policy Name**, set the Stream as Incoming Request, and click **Add**.
12. Replace the existing default content in the Body section with the following policy XML, and click **Update**.

Condition String:

```
(request.verb = "POST" OR request.verb = "PUT" OR request.verb = "DELETE")
```

Policy Message:

```
<!-- This policy can be used to create or modify the standard HTTP request and response messages -->
<AssignMessage async="false" continueOnError="false" enabled="true" xmlns='http://www.sap.com/apimgmt'>
  <!-- Sets a new value to the existing parameter -->
  <Set>
    <Headers>
      <Header name="x-csrf-token">{callOutResponse.header.x-csrf-token}</Header>
      <Header
        name="Cookie">{callOutResponse.header.Set-Cookie.1};{callOutResponse.header.Set-Cookie.2};
{callOutResponse.header.Set-Cookie.3}</Header>
    </Headers>
  </Set>
  <IgnoreUnresolvedVariables>false</IgnoreUnresolvedVariables>
  <AssignTo createNew="false" type="request">request</AssignTo>
</AssignMessage>
```



Note:

Some customers may receive only two cookies instead of three. In such cases, update the policy accordingly.

13. Click **Save**.

4. Connect to Maximo

Configure IBM Maximo Application Suite 9 to publish asset, location, and work order changes to iMaintenance through Kafka.

Complete this chapter if iMaintenance connects to Maximo. If it connects to SAP, see [Connect to SAP \(on page 14\)](#). When this chapter is complete, continue with [Initialize iMaintenance \(on page 157\)](#).

The integration has two directions:

- **Maximo to iMaintenance:** Maximo publish channels send changes through Maximo queues to Kafka topics, which iMaintenance reads. Maximo connects to Kafka with the Kafka user and certificate supplied by the Innovapptive infrastructure team.
- **iMaintenance to Maximo:** iMaintenance calls the Maximo API with the API key of the Maximo integration user.

The Kafka credentials and the Maximo API key are different credentials with different purposes.

Do the tasks in this order. Values in angle brackets, such as `<site-id>`, are deployment inputs.

1. [Check Maximo Prerequisites \(on page 53\)](#)
2. [Set Up Maximo Integration Components \(on page 54\)](#)
3. [Connect iMaintenance to Maximo \(on page 155\)](#)
4. [Expose and Secure Maximo APIs \(on page 155\)](#)
5. [Verify the Maximo Connection \(on page 156\)](#)

4.1. Check Maximo Prerequisites

Check MAS 9 integration access and collect the Kafka settings and site values needed for the deployment.

- The Integration Framework is enabled in the MAS 9 environment.
- The Maximo team has administrative access to Maximo (for example, `MAXADMIN`) and to MAS Suite administration.
- Network access is planned: Maximo reaches the Kafka bootstrap server and port over TLS, and iMaintenance reaches the Maximo base URL over HTTPS. See [Connect iMaintenance to Maximo \(on page 155\)](#).
- The Innovapptive infrastructure team has supplied the Kafka details for this environment: bootstrap server and port, TLS certificate, user name, and password.

Deployment inputs

Input	Used in
<code><site-id></code> : the sites allowed to send data	Integration user default site, <code>INNOSKIPSITE</code> , security group sites

Input	Used in
Integration user name (default <code>INNOVAPPTIVE</code>)	Integration user, processing rules
<code><bootstrap-server>:<port></code> , <code><kafka-username></code> , <code><kafka-password></code> , certificate	Endpoints, message provider
<code><location></code> and <code><location-prefix></code> values to exclude	Integration controls, location filter queries, <code>INNO_3M_SKIP2</code> rules

4.2. Set Up Maximo Integration Components

Create the Maximo objects, event channels, and Kafka configuration used by the iMaintenance integration.

Do these tasks in the order listed. Later tasks use the objects created earlier.

1. [Create Maximo Custom Attributes \(on page 54\)](#)
2. [Create Maximo Relationships \(on page 56\)](#)
3. [Create the Maximo Condition \(on page 58\)](#)
4. [Create the Maximo Integration User \(on page 58\)](#)
5. [Create the Integration Object Structures \(on page 60\)](#)
6. [Configure Object Structure Queries \(on page 97\)](#)
7. [Create Integration Controls \(on page 99\)](#)
8. [Create the iMaintenance Publish Channels \(on page 100\)](#)
9. [Configure Maximo Kafka Endpoints \(on page 132\)](#)
10. [Configure the Kafka Message Provider \(on page 134\)](#)
11. [Create the Maximo Queues \(on page 137\)](#)
12. [Create the Maximo Automation Scripts \(on page 137\)](#)
13. [Configure Maximo Roles and Authorizations \(on page 146\)](#)
14. [Activate Maximo Kafka Cron Tasks \(on page 149\)](#)
15. [Configure the iMaintenance External System \(on page 152\)](#)

4.2.1. Create Maximo Custom Attributes

Add the custom attributes that the iMaintenance integration objects and scripts use, and apply the database configuration.

`LM_WORKORDERID` on `ASSIGNMENT` is written by the `INNO_ASSIGNMENT` automation script. `INNO_CHANGEDATE` is written by the `INNO_CHANGEDATE` script.

1. In the **Database Configuration** application, add the following attributes to their objects.

OBJECT- NAME	AT- TRIBUTE- NAME	TITLE	REMARKS	MAXTYPE	LENGTH	SEARCHTYPE	LOCALIZ- ABLE
TOOLINV	INNO_- CHANGE- DATE3	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
WORKO- RDER	INNO_- WOPRIORI- TY	Priority	Priority	ALN	25	NONE	1
INVBAL- ANCES	INVBAL- ANCES	Invbalances	Invbalances	ALN	50	NONE	1
ASSIGN- MENT	LM_- WORKO- RDERID	Workorder ID for as- signments	Workorder ID	BIGINT	19	EXACT	0
INVENTORY	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
SKDEX- TRACAP- TOOLVIEW	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
TOOLITEM	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
SERVICEITEMS	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
TOOLINV	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
ITEM	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
LABOR	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
PERSON	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
WPSERVICE	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
WPTOOL	INNO_- CHANGE- DATE	Inno_- changedate	Inno_- changedate	DATETIME	10	EXACT	0
WP-	INNO -	Inno -	Inno -	DATETIME	10	EXACT	0

2. Apply the configuration changes.

Result

The attributes exist on their objects and the database configuration is applied.

4.2.2. Create Maximo Relationships

Create the database relationships that the integration object structures and processing rules use.

INNO_LOCATION is created on three parent objects (DOCLINKS, ASSIGNMENT, and WORKLOG), each with its own where clause.

In the **Database Configuration** application, open each parent object, and on the **Relationships** tab add the relationships below with their where clauses.

Name	Parent object	Child object
INNO_SECMGR	GROUPUSER	APPLICATIONAUTH
INNO_APIKEY	MAXUSER	APIKEYTOKEN
INNO_CLASSUSEWITH	CLASSSTRUCTURE	CLASSUSEWITH
INNO_CLASSSPECUSEWITH	CLASSSPEC	CLASSSPECUSEWITH
INNO_NUMERICDOMAIN	CLASSSPEC	NUMERICDOMAIN
INNOV_ALNDOMAIN	CLASSSPEC	ALNDOMAIN
INNO_SITE	DOCLINKS	SITE
INNO_LOCATION	DOCLINKS	LOCATIONS
INNO_LOCATION	ASSIGNMENT	LOCATIONS
INNO_LOCATION	WORKLOG	LOCATIONS

INNO_SECMGR (GROUPUSER to APPLICATIONAUTH)

```
groupname=:groupname AND (optionname IN
('READ','SAVE','INSERT','RELATERECORD','MODELWL','MANAGEFOLD','ASSOCFOLD','MANAGELIB','CREATEWO','STATUS')
and app in ('SR','ASSET','CREATEDR','RECEIPTS','CREATESR','VIEWSR'))
or (app in ('WOTRACK','QUICKREP') and optionname in('OWNERSHIP',
'READ','INSERT','SAVE','CREATEWO','MANAGEFOLD','ASSOCFOLD','MANAGELIB','STATUS','RELATERECORD','REPDOWN','E
SCHDASSW','CREATESREQ'))
```

INNO_APIKEY (MAXUSER to APIKEYTOKEN)

```
userid=:userid
```

INNO_CLASSUSEWITH (CLASSSTRUCTURE to CLASSUSEWITH)

```
classstructureid=:classstructureid and OBJECTVALUE IN ('SR','WORKORDER')
```

INNO_CLASSSPECUSEWITH (CLASSSPEC to CLASSSPECUSEWITH)

```
classspecid=:classspecid and OBJECTVALUE IN ('SR','WORKORDER')
```

INNO_NUMERICDOMAIN (CLASSSPEC to NUMERICDOMAIN)

```
domainid=:domainid
```

INNOV_ALNDOMAIN (CLASSSPEC to ALNDOMAIN)

```
domainid=:domainid
```

INNO_SITE (DOCLINKS to SITE)

4.2.3. Create the Maximo Condition

Create the C_WAPPR condition used by the integration.

In the **Conditional Expression Manager** application, create the following condition.

CONDITIONNUM	DESCRIPTION	Type	Expression	Always Evaluate
C_WAPPR	If WO Status is WAPPR	Expression	:status<>'APPR'	yes

4.2.4. Create the Maximo Integration User

Create the integration user in MAS, create its Maximo API key, and set its default insert site.

You can use another name for this account. If you do, use the same name in every processing rule that compares an account with **INNOVAPTIVE** :

- **INNO_ATTACHMENT** , rule **INNO_SKIP2** : **CREATEBY**
- **INNO_LABTRANS** , rule **INNO_SKIP2** : **ENTERBY**
- **INNO_WO** and **INNO_WOACT** , rule **INNO_SKIP2** : **CHANGEBY**
- **INNV_WORKLOG** , rule **INNO_SKIP2** : **CREATEBY**

1. In MAS Suite administration, create the following user.

Field	Value
User	INNOVAPPTIVE
Person	INNOVAPPTIVE
Status	ACTIVE
Display name	INNOVAPPTIVE API User

2. Wait until the user is synchronized to Maximo Manage.

Figure 4-1 The integration user INNOVAPPTIVE in the Maximo Users application (e-mail address masked)

The screenshot displays the 'User' profile for 'INNOVAPPTIVE' in the Maximo Users application. The interface includes tabs for 'User', 'Groups', and 'Security Profile'. The 'User' tab is active, showing fields for 'User' (INNOVAPPTIVE), 'Status' (ACTIVE), and 'Type' (TYPE 1). Below this is the 'Login Information' section with 'User Name' (innovapptive) and a 'Set Password' button. The 'Personal' section contains fields for 'Person' (INNOVAPPTIVE), 'Display Name' (innovapptive), 'Address' (masked), 'Status' (ACTIVE), 'Primary Phone' (masked), 'City' (masked), 'First Name' (masked), 'Primary E-mail' (masked), 'State/Province' (masked), 'Last Name' (masked), 'Workflow Delegate' (masked), 'ZIP/Postal Code' (masked), 'Supervisor' (masked), 'Memo' (masked), and 'Country' (masked).

3. In Maximo, create an API key for this user.
4. In the **Users** application, set the default insert site of the user to `<site-id>`.

Next action

You add this user to the `INNO_MAXUSER` security group later, after the group is created. See [Configure Maximo Roles and Authorizations \(on page 146\)](#).

4.2.5. Create the Integration Object Structures

Create the twelve object structures that the integration publishes through and that iMaintenance reads from Maximo.

Create the object structures in the order listed. Some are published through publish channels; others are read through the Maximo API.

- [Create INNO_ASSET \(on page 60\)](#)
- [Create INNO_ASSIGNMENT \(on page 64\)](#)
- [Create INNO_ATTACHMENT \(on page 66\)](#)
- [Create INNV_WORKLOG \(on page 69\)](#)
- [Create INNO_CLASSIFICATION \(on page 71\)](#)
- [Create INNO_GETSR \(on page 74\)](#)
- [Create INNO_GETWO \(on page 78\)](#)
- [Create INNO_INVBAL \(on page 83\)](#)
- [Create INNO_LABTRANS \(on page 86\)](#)
- [Create INNO_LOCATION \(on page 89\)](#)
- [Create INNO_USER \(on page 92\)](#)
- [Create INNO_RELATEDRECORD \(on page 95\)](#)

4.2.5.1. Create INNO_ASSET

Create the INNO_ASSET object structure (Custom Maximo API for Asset).

1. In the **Object Structures** application, duplicate **MXAPIASSET** and enter the following values.

Field	Value
Object Structure	INNO_ASSET
DESCRIPTION	Custom Maximo API for Asset
Consumed By	INTEGRATION
Authorization Name	MXAPIASSET
Outbound Definition Class	psdi.iface.app.asset.MoutAssetProcess
Inbound Processing Class	psdi.iface.app.asset.MaxAssetProcess
Search Attributes	
Data restriction condition	
Module Name	ASSET
Auto Paging Threshold	-1
Primary API?	1
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	1

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	Exclude Attributes Mode
ASSET		ASSET		1	N
ASSETMETER	ASSET	ASSET/ASSET-METER	INT_ASSET-METER	1	N
ASSETSPEC	ASSET	ASSET/ASSETSPEC	ASSETSPEC-CLASS	3	N
DOCLINKS	ASSET	ASSET/DOCLINKS	DOCLINKS	4	N
DOWNTIMEREPORT	ASSET	ASSET/DOWNTIMEREPORT	DOWNTIMEREPORT	6	N
MODDOWNTIMEHIST	ASSET	ASSET/MODDOWNTIMEHIST	MODDOWNTIMEHIST	7	N

Figure 4-2 INNO_ASSET object structure with its source objects

Object Structure

Consumed by: INTEGRATION

Integration Application: _____

Authorization Name: MXAPIASSET

Outbound Definition Class: pdi.iface.app.asset.MoutAHAssetProcess

Inbound Processing Class: pdi.iface.app.asset.MaxAssetProcess

Search Attributes: _____

Data restriction condition: _____

Module Name: ASSET

Auto Paging Threshold: -1

Self Referencer: ☐

Commutator: ☒

Publish Complete Hierarchy: ☐

Alias Conflict: ☐

Support Flat Structure: ☐

Load Query From All Apps: ☒

Internal: ☐

Source Objects for INNO_ASSET (1 - 6 of 6)

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?	Exclude Attributes Mode?	
ASSET		ASSET		1	☒	☐	
ASSETMETER	ASSET	ASSET/ASSETMETER	INT_ASSETMETER	1	☒	☐	
ASSETSPEC	ASSET	ASSET/ASSETSPEC	ASSETSPECCLASS	3	☒	☐	
DOCLINKS	ASSET	ASSET/DOCLINKS	DOCLINKS	4	☒	☐	
DOWNTIMEREPORT	ASSET	ASSET/DOWNTIMEREPORT	DOWNTIMEREPORT	6	☒	☐	
MODDOWNTIMEHIST	ASSET	ASSET/MODDOWNTIMEHIST	MODDOWNTIMEHIST	7	☒	☐	

3. Set the following field states (included, excluded, restricted, and non-persistent fields).

INTOBJECTNAME	OBJECTNAME	NAME	INTOBJFLDTYPE
INNO_ASSET	ASSET	AGE	NONPERSISTENT
INNO_ASSET	ASSET	AHASSETFLAGS	NONPERSISTENT
INNO_ASSET	ASSET	AHHASPLAN	NONPERSISTENT
INNO_ASSET	ASSET	AHHEALTHTRENDIRECTION	NONPERSISTENT
INNO_ASSET	ASSET	AHHEALTHTRENDVALUE	NONPERSISTENT
INNO_ASSET	ASSET	AHMRR	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCAPEXYEAR1	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCAPEXYEAR2	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCAPEXYEAR3	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCAPEXYEAR4	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCAPEXYEAR5	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCOSTYEAR1	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCOSTYEAR2	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCOSTYEAR3	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCOSTYEAR4	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANCOSTYEAR5	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANLIFEYEAR1	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANLIFEYEAR2	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANLIFEYEAR3	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANLIFEYEAR4	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANLIFEYEAR5	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANOPEXYEAR1	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANOPEXYEAR2	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANOPEXYEAR3	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANOPEXYEAR4	NONPERSISTENT
INNO_ASSET	ASSET	AHPLANOPEXYEAR5	NONPERSISTENT
INNO_ASSET	ASSET	AHRULIFE	NONPERSISTENT
INNO_ASSET	ASSET	ASSETUID	NONRESTRICTED
INNO_ASSET	ASSET	BINNUM	RESTRICTED
INNO_ASSET	ASSET	CHANGEPMSTATUS	NONPERSISTENT
INNO_ASSET	ASSET	CLASSSTRUCTUREID	RESTRICTED
INNO_ASSET	ASSET	DESCRIPTION_ - LONGDESCRIPTION	NONPERSISTENT
INNO_ASSET	ASSET	GEOMETRY	NONPERSISTENT
INNO_ASSET	ASSET	EXCLUDE	EXCLUDE

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
ASSET	N	N	N	N	N	N
ASSETMETER	Y	Y	N	N	Y	N
ASSETSPEC	Y	Y	N	N	Y	N
DOCLINKS	Y	Y	N	N	Y	N
DOWN-TIMEREPOR	Y	Y	N	N	Y	Y
MODDOWN-TIMEHIST	Y	Y	N	N	Y	Y

4.2.5.2. Create INNO_ASSIGNMENT

Create the INNO_ASSIGNMENT object structure (Custom OS to Send Assignment).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_ASSIGNMENT
DESCRIPTION	Custom OS to Send Assignment
Consumed By	INTEGRATION
Authorization Name	MXAPIASSIGNMENT
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

Figure 4-3 INNO_ASSIGNMENT object structure definition

Object Structure

Object Structure

INNO_ASSIGNMENT

Custom OS to Send Assignment

Consumed By

INTEGRATION

Integration Application

Authorization Name

MXAPIASSIGNMENT

Outbound Definition Class

Inbound Processing Class

Search Attributes

Data restriction condition

Module Name

Auto Paging Threshold

-1

Primary API?

User Defined?

Self Reference?

Configurable?

Publish Complete Hierarchy?

Alias Conflict?

Support Flat Structure?

Load Query From All Apps?

Internal?

2. Add **ASSIGNMENT** as the source object (object order 1, no parent object or relationship).

Figure 4-4 ASSIGNMENT as the source object of INNO_ASSIGNMENT

Source Objects for INNO_ASSIGNMENT (1 - 1 of 1)

Object	Parent Object	Object Location Path	Relationship	Object Order
ASSIGNMENT		ASSIGNMENT		1

Details

Object: ASSIGNMENT

Parent Object:

Object Location Path: ASSIGNMENT

Exclude Attributes Mode?: ☐

Relationship:

Object Order: 1

Alternate Key:

License:

User Defined?: ☒

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.
4. From the **Select Action** menu, choose **Configure Object Structure Security**, and select **MXAPIASSIGNMENT** as the authorization name.

Figure 4-5 Configure Object Structure Security for INNO_ASSIGNMENT

Object Structure

Object Structure: INNO_ASSIGNMENT

Authorization Name: MXAPIASSIGNMENT

Use Object Structure for Authorization Name?: ☒

Custom OS to Send Assignment

UI Assignment Definition

Add/Modify Signature Options (1 - 4 of 4)

Option	Description	Prerequisite	Visible?
DELETE	Delete MXAPIASSIGNMENT	SAVE	☒
INSERT	Insert MXAPIASSIGNMENT	SAVE	☒
READ	Read MXAPIASSIGNMENT		☒
SAVE	Save MXAPIASSIGNMENT		☒

Select Service Option

Cancel OK

4.2.5.3. Create INNO_ATTACHMENT

Create the INNO_ATTACHMENT object structure (INNOVAPPTIVE custom API to send Doclinks).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_ATTACHMENT
DESCRIPTION	INNOVAPPTIVE custom API to send Doclinks
Consumed By	INTEGRATION
Authorization Name	INNO_ATTACHMENT
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

Figure 4-6 INNO_ATTACHMENT object structure definition

The screenshot shows the 'Object Structure' form for 'INNO_ATTACHMENT'. The form is divided into two main sections: a left column for text-based fields and a right column for toggle-based settings.

Left Column Fields:

- Object Structure:** A dropdown menu showing 'INNO_ATTACHMENT'.
- DESCRIPTION:** A text field containing 'INNOVAPPTIVE custom API to send Doclinks'.
- Consumed By:** A dropdown menu showing 'INTEGRATION'.
- Authorization Name:** A text field containing 'INNO_ATTACHMENT'.
- Outbound Definition Class:** An empty text field.
- Inbound Processing Class:** An empty text field.
- Search Attributes:** An empty text field.
- Data restriction condition:** An empty text field.
- Module Name:** An empty text field.
- Auto Paging Threshold:** A text field containing '-1'.

Right Column Toggles:

- Primary API?:** A toggle switch, currently off.
- User Defined?:** A toggle switch, currently on (checked).
- Self Reference?:** A toggle switch, currently off.
- Configurable?:** A toggle switch, currently on (checked).
- Publish Complete Hierarchy?:** A toggle switch, currently off.
- Alias Conflict?:** A toggle switch, currently off.
- Support Flat Structure?:** A toggle switch, currently off.
- Load Query From All Apps?:** A toggle switch, currently off.
- Internal?:** A toggle switch, currently off.

2. Add **DOCLINKS** as the source object (object order 1, no parent object or relationship).

Figure 4-7 DOCLINKS as the source object of INNO_ATTACHMENT

Source Objects for INNO_ATTACHMENT (1 - 1 of 1)

Object	Parent Object	Object Location Path	Relationship	Object Order
DOCLINKS		DOCLINKS		1

Details

Object: DOCLINKS

Parent Object:

Object Location Path: DOCLINKS

Exclude Attributes Mode? ☐

Relationship:

Object Order: 1

Alternate Key:

License:

User Defined? ☒

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.
4. From the **Select Action** menu, choose **Configure Object Structure Security**, and select the **Use Object Structure for Authorization Name** check box.

Figure 4-8 Configure Object Structure Security for INNO_ATTACHMENT

Object Structure

Configure Object Structure Security

Object Structure: INNO_ATTACHMENT

Authorization Name: INNO_ATTACHMENT

Use Object Structure for Authorization Name? ☒

Add/Modify Signature Options (1 - 5 of 5)

Option	Description	Prerequisite	Visible?
BMXVIEWMANAGEWI	Sigoption to control the visibility of SQL clause	READ	☒
DELETE	Delete INNO_ATTACHMENT	SAVE	☒
INSERT	Insert INNO_ATTACHMENT	SAVE	☒
READ	Read INNO_ATTACHMENT		☒
SAVE	Save INNO_ATTACHMENT		☒

1 - 5 of 5

Cancel OK

4.2.5.4. Create INNV_WORKLOG

Create the INNV_WORKLOG object structure (Custom OS to send the Work Log update to iMaintenance).

1. In the **Object Structures** application, duplicate **MXAPIWORKLOG** and enter the following values.

Field	Value
Object Structure	INNV_WORKLOG
DESCRIPTION	Custom OS to send the Work Log update to iMaintenance
Consumed By	INTEGRATION
Authorization Name	MXAPIWORKLOG
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

INTOBJECTNAME	OBJECTNAME	NAME	INTOBJFLDTYPE
INNV_WORKLOG	WORKLOG	DESCRIPTION_- LONGDESCRIPTION	NONPERSISTENT
INNV_WORKLOG	WORKLOG	HASLD	EXCLUDE
INNV_WORKLOG	WORKLOG	LANGCODE	EXCLUDE

Figure 4-9 INNV_WORKLOG object structure definition

Object Structure

Object Structure

INNV_WORKLOG

Custom OS to send the Work Log update to iMaintenance

Consumed By

INTEGRATION

Integration Application

Authorization Name

MXAPIWORKLOG

Outbound Definition Class

Inbound Processing Class

Search Attributes

Data restriction condition

Module Name

Primary API?

Self Reference?

Publish Complete Hierarchy?

Support Flat Structure?

User Defined?

Configurable?

Alias Conflict?

Load Query From All Apps?

Internal?

2. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.
3. From the **Select Action** menu, choose **Configure Object Structure Security**, and select **MXAPIWORKLOG** as the authorization name.

4.2.5.5. Create INNO_CLASSIFICATION

Create the INNO_CLASSIFICATION object structure, which gives access to classifications, their specifications, and domains.

1. In the **Object Structures** application, duplicate **MXAPICLASSTRUCTURE** and enter the following values.

Field	Value
Object Structure	INNO_CLASSIFICATION
DESCRIPTION	Innovapptive Classification
Consumed By	INTEGRATION
Authorization Name	MXAPICLASSTRUCTURE
Outbound Definition Class	
Inbound Processing Class	psdi.iface.app.classify.MaxClassificationProcess
Search Attributes	
Data restriction condition	
Module Name	SETUP
Auto Paging Threshold	-1
Primary API?	1
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	Exclude Attributes Mode
CLASSSTRUCTURE		CLASSSTRUCTURE		1	N
CLASSSPEC	CLASSSTRUCTURE	CLASSSTRUCTURE/CLASSSPEC	CLASSSPEC	1	N
CLASSSPECUSEWITH	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/CLASSSPECUSEWITH	INNO_CLASSSPECUSEWITH	1	N
ASSETATTRIBUTE	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/ASSETATTRIBUTE	ASSETATTRIBUTE	2	N
ALNDOMAIN	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/ALNDOMAIN	INNOV_ALNDOMAIN	3	N
NUMERICDOMAIN	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/NUMERICDOMAIN	INNO_NUMERICDOMAIN	4	N
CLASSUSEWITH	CLASSSTRUCTURE	CLASSSTRUCTURE/CLASSUSEWITH	INNO_CLASSUSEWITH	2	N
CLASSANCESFOR	CLASSSTRUCTURE	CLASSSTRUCTURE/CLASSANCESFOR	CLASSANCESFOR	3	N

Figure 4-10 INNO_CLASSIFICATION object structure with its eight source objects

Object Structure

Object Structure

INNO_CLASSIFI

Innovapptive Classification

Consumed By

INTEGRATION

Integration Application

Authorization Name

MXAPICLASSSTRUCTURE

Outbound Definition Class

Inbound Processing Class

pedi.face.app.classify.MaxClassificationProcess

Search Attributes

Data restriction condition

Module Name

SETUP

Auto Paging Threshold

-1

Primary API?

☒

Self Reference?

☐

Publish Complete Hierarchy?

☐

Support Flat Structure?

☐

User Defined?

☐

Configurable?

☐

Alias Conflict?

☐

Load Query From All Apps?

☐

Internal?

☐

Source Objects for INNO_CLASSIFICATION (1 - 8 of 8)

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?	Exclude Attributes Mode?	
CLASSSTRUCTURE		CLASSSTRUCTURE		1	☒	☐	
CLASSSPEC	CLASSSTRUCTURE	CLASSSTRUCTURE/CLASSSPEC	CLASSSPEC	1	☒	☐	
CLASSSPECUSEWITH	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/CLASSSPECUSEWITH	INNO_CLASSSPECUSEWITH	1	☒	☐	
ASSETATTRIBUTE	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/ASSETATTRIBUTE	ASSETATTRIBUTE	2	☒	☐	
ALNDOMAIN	CLASSSPEC	CLASSSTRUCTURE/CLASSSPEC/ALNDOMAIN	INNOV_ALNDOMAIN	3	☒	☐	

3. Set the following field states (included, excluded, restricted, and non-persistent fields).

INTOBJECTNAME	OBJECTNAME	NAME	INTOBJFLDTYPE
INNO_CLASSIFICATION	CLASSSPEC	DATATYPE	NONPERSISTENT
INNO_CLASSIFICATION	CLASSSTRUCTURE	CLASSSTRUCTUREUID	EXCLUDE
INNO_CLASSIFICATION	CLASSSTRUCTURE	DESCRIPTION_ LONGDESCRIPTION	NONPERSISTENT
INNO_CLASSIFICATION	CLASSSTRUCTURE	HASCHILDREN	EXCLUDE
INNO_CLASSIFICATION	CLASSSTRUCTURE	HASLD	EXCLUDE
INNO_CLASSIFICATION	CLASSSTRUCTURE	HIERARCHYPATH	NONPERSISTENT
INNO_CLASSIFICATION	CLASSSTRUCTURE	LANGCODE	EXCLUDE
INNO_CLASSIFICATION	CLASSSTRUCTURE	OBJECTNAME	NONPERSISTENT
INNO_CLASSIFICATION	CLASSSTRUCTURE	PARENT	RESTRICTED

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
CLASSSTRUCTURE	N	N	N	N	N	N
CLASSSPEC	Y	Y	N	N	Y	N
CLASSSPECUSEWITH	Y	Y	N	N	Y	N
ASSETATTRIBUTE	Y	Y	N	N	Y	N
ALNDOMAIN	Y	Y	N	N	Y	N
NUMERICDOMAIN	Y	Y	N	N	Y	N
CLASSUSEWITH	Y	Y	N	N	Y	N
CLASSANCES-TOR	Y	Y	N	N	Y	N

5. Create the OSLC queries `classificationasset`, `classificationrs`, and `classificationwo` on this object structure. See [Configure Object Structure Queries \(on page 97\)](#).

4.2.5.6. Create INNO_GETSR

Create the INNO_GETSR object structure (Innovapptive Custom OS for Service request).

1. In the **Object Structures** application, duplicate **MXAPISR** and enter the following values.

Field	Value
Object Structure	INNO_GETSR
DESCRIPTION	Innovapptive Custom OS for Service request
Consumed By	INTEGRATION
Authorization Name	MXAPISR
Outbound Definition Class	
Inbound Processing Class	psdi.iface.app.ticket.MaxSRProcess
Search Attributes	
Data restriction condition	
Module Name	SD
Auto Paging Threshold	-1
Primary API?	1
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	1

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	Exclude Attributes Mode
SR		SR		1	N
DOCLINKS	SR	SR/DOCLINKS	DOCLINKS	1	N
WORKLOG	SR	SR/WORKLOG	UXWORKLOG	2	N
RELATE-DRECORD	SR	SR/RELATE-DRECORD	ONLYRELATE-DRECORD	3	N

Figure 4-11 INNO_GETSR object structure with its source objects

Object Structure

Object Structure
INNO_GETSR
Innovapptive Custom OS for Service request

Consumed By
INTEGRATION
Integration Application

Authorization Name
MXAPISR

Outbound Definition Class

Inbound Processing Class
pedi.iface.app.ticket.MaxSRProcess

Search Attributes

Data restriction condition

Module Name
SD

Auto Paging Threshold
-1

Primary API? ☒

User Defined? ☒

Self Reference? ☐

Configurable? ☒

Publish Complete Hierarchy? ☐

Alias Conflict? ☐

Support Flat Structure? ☐

Load Query From All Apps? ☒

Internal? ☐

Source Objects for INNO_GETSR (1 - 4 of 4)

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?	Exclude Attributes Mode?
SR		SR		1	☒	☐
DOCLINKS	SR	SR/DOCLINKS	DOCLINKS	1	☒	☐
WORKLOG	SR	SR/WORKLOG	UXWORKLOG	2	☒	☐
RELATEDRECORD	SR	SR/RELATEDRECORD	ONLYRELATEDRECORD	3	☒	☐

1 - 4 of 4

3. Set the following field states (included, excluded, restricted, and non-persistent fields).

INTOBJECTNAME	OBJECTNAME	NAME	INTOBJFLDTYPE
INNO_GETSR	DOCLINKS	ADDINFO	NONPERSISTENT
INNO_GETSR	DOCLINKS	APP	NONPERSISTENT
INNO_GETSR	DOCLINKS	DESCRIPTION	NONPERSISTENT
INNO_GETSR	DOCLINKS	DMSNAME	NONPERSISTENT
INNO_GETSR	DOCLINKS	DOCUMENTDATA	NONPERSISTENT
INNO_GETSR	DOCLINKS	KEYCOLUMN	NONPERSISTENT
INNO_GETSR	DOCLINKS	NEWURLNAME	NONPERSISTENT
INNO_GETSR	DOCLINKS	SHOW	NONPERSISTENT
INNO_GETSR	DOCLINKS	UPLOAD	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLNAME	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLPARAM1	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLPARAM2	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLPARAM3	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLPARAM4	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLPARAM5	NONPERSISTENT
INNO_GETSR	DOCLINKS	URLTYPE	NONPERSISTENT
INNO_GETSR	DOCLINKS	WEBURL	NONPERSISTENT
INNO_GETSR	SR	DESCRIPTION_- LONGDESCRIPTION	NONPERSISTENT
INNO_GETSR	SR	FR1CODE_LONGDESCRIP- TION	NONPERSISTENT
INNO_GETSR	SR	FR2CODE_LONGDESCRIP- TION	NONPERSISTENT
INNO_GETSR	SR	GEOMETRY	NONPERSISTENT
INNO_GETSR	SR	HASLD	EXCLUDE
INNO_GETSR	SR	LANGCODE	EXCLUDE
INNO_GETSR	SR	NP_STATUSMEMO	NONPERSISTENT
INNO_GETSR	SR	PROBLEMCODE_- LONGDESCRIPTION	NONPERSISTENT
INNO_GETSR	SR	REMARKDESC_- LONGDESCRIPTION	NONPERSISTENT
INNO_GETSR	SR	REPORTEDBYNAME	NONPERSISTENT
INNO_GETSR	TKSERVICEADDRESS	CITY	NONPERSISTENT
INNO_GETSR	TKSERVICEADDRESS	CLASS	NONPERSISTENT
INNO_GETSR	TKSERVICEADDRESS	FORMATTEDADDRESS	NONPERSISTENT
INNO_GETSR	TKSERVICEADDRESS	HASLD	EXCLUDE

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
SR	N	N	N	N	N	N
DOCLINKS	Y	Y	N	N	Y	N
WORKLOG	Y	Y	N	N	Y	N
RELATE-DRECORD	Y	Y	N	N	Y	N

4.2.5.7. Create INNO_GETWO

Create the INNO_GETWO object structure (WO Details API).

1. In the **Object Structures** application, duplicate **MXAPIWODETAIL** and enter the following values. Then add the additional objects below.

Field	Value
Object Structure	INNO_GETWO
DESCRIPTION	WO Details API
Consumed By	INTEGRATION
Authorization Name	MXAPIWODETAIL
Outbound Definition Class	psdi.iface.app.wo.MoutWORsvProcess
Inbound Processing Class	psdi.iface.mic.StatefulMicSetIn
Search Attributes	
Data restriction condition	
Module Name	WO
Auto Paging Threshold	
Primary API?	1
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	1

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	Exclude Attributes Mode
WORKORDER		WORKORDER		1	N
WPLABOR	WORKORDER	WORKORDER/WPLABOR	SHOWPLANLABOR	1	N
WPMATERIAL	WORKORDER	WORKORDER/WPMATERIAL	SHOWPLANMATERIAL	2	N
WPTOOL	WORKORDER	WORKORDER/WPTOOL	SHOWPLANTOOL	3	N
WPSERVICE	WORKORDER	WORKORDER/WPSERVICE	SHOWPLANSERVICE	4	N
INVRESERVE	WORKORDER	WORKORDER/INVRESERVE	MXINTINVRES	5	N
DOCLINKS	WORKORDER	WORKORDER/DOCLINKS	DOCLINKS	6	N
WORKLOG	WORKORDER	WORKORDER/WORKLOG	WOWORKLOG	7	N
MATUSETRANS	WORKORDER	WORKORDER/MATUSETRANS	UXSHOWACTUALMATERIAL	9	N
LABTRANS	WORKORDER	WORKORDER/LABTRANS	UXSHOWACTUALLABOR	10	N
TOOLTRANS	WORKORDER	WORKORDER/TOOLTRANS	UXSHOWACTUALTOOL	11	N
WOACTIVITY	WORKORDER	WORKORDER/WOACTIVITY	WOACTIVITY	12	N
DOCLINKS	WOACTIVITY	WORKORDER/WOACTIVITY/DOCLINKS	DOCLINKS	19	N
RELATERECORD	WORKORDER	WORKORDER/RELATERECORD	RELATERECORD	13	N
FAILUREREPORT	WORKORDER	WORKORDER/FAILUREREPORT	UXSHOWFAILUREREPORT	14	N
MODDOWNTIMEHIST	WORKORDER	WORKORDER/MODDOWNTIMEHIST	MODDOWNTIMEHIST	16	N
ASSIGNMENT	WORKORDER	WORKORDER/ASSIGNMENT	SHOWASSIGNMENT	17	N

3. Set the following field states (included, excluded, restricted, and non-persistent fields).

INTOBJECTNAME	OBJECTNAME	NAME	INTOBJFLDTYPE
INNO_GETWO	ASSIGNMENT	TASKID	NONPERSISTENT
INNO_GETWO	CLASSSTRUCTURE	HASLD	EXCLUDE
INNO_GETWO	CLASSSTRUCTURE	LANGCODE	EXCLUDE
INNO_GETWO	DOCLINKS	ADDINFO	NONPERSISTENT
INNO_GETWO	DOCLINKS	APP	NONPERSISTENT
INNO_GETWO	DOCLINKS	DESCRIPTION	NONPERSISTENT
INNO_GETWO	DOCLINKS	DMSNAME	NONPERSISTENT
INNO_GETWO	DOCLINKS	DOCUMENTDATA	NONPERSISTENT
INNO_GETWO	DOCLINKS	KEYCOLUMN	NONPERSISTENT
INNO_GETWO	DOCLINKS	NEWURLNAME	NONPERSISTENT
INNO_GETWO	DOCLINKS	SHOW	NONPERSISTENT
INNO_GETWO	DOCLINKS	UPLOAD	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLNAME	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLPARAM1	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLPARAM2	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLPARAM3	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLPARAM4	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLPARAM5	NONPERSISTENT
INNO_GETWO	DOCLINKS	URLTYPE	NONPERSISTENT
INNO_GETWO	DOCLINKS	WEBURL	NONPERSISTENT
INNO_GETWO	INVRESERVE	DESCRIPTION_- LONGDESCRIPTION	NONPERSISTENT
INNO_GETWO	INVRESERVE	HASLD	EXCLUDE
INNO_GETWO	INVRESERVE	LANGCODE	EXCLUDE
INNO_GETWO	LABTRANS	ACTUALTASKID	NONPERSISTENT
INNO_GETWO	LABTRANS	FINISHDATE	NONPERSISTENT
INNO_GETWO	LABTRANS	FINISHTIME	NONPERSISTENT
INNO_GETWO	LABTRANS	STARTDATE	NONPERSISTENT
INNO_GETWO	LABTRANS	STARTTIME	NONPERSISTENT
INNO_GETWO	LABTRANS	TASKID	NONPERSISTENT
INNO_GETWO	MATUSETRANS	ACTUALTASKID	NONPERSISTENT
INNO_GETWO	MATUSETRANS	ITEMNUM	NONRESTRICTED
INNO_GETWO	MATUSETRANS	POSITIVEQUANTITY	NONPERSISTENT
INNO_GETWO	MATUSETRANS	REQUESTNUM	NONPERSISTENT
INNO_GETWO	MATUSETRANS	SITEID	NONRESTRICTED

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
WORKORDER	N	N	N	N	N	N
WPLABOR	Y	Y	N	N	Y	N
WPMATERIAL	N	Y	N	N	Y	N
WPTOOL	Y	Y	N	N	Y	N
WPSERVICE	Y	Y	N	N	Y	N
INVRESERVE	Y	Y	N	N	Y	N
WORKLOG	Y	Y	N	N	Y	N
MATUSE-TRANS	Y	Y	N	N	Y	N
LABTRANS	Y	Y	N	N	Y	N
TOOLTRANS	Y	Y	N	N	Y	N
WOACTIVITY	N	Y	N	N	Y	N
RELATE-DRECORD	Y	Y	N	N	Y	N
FAILUREREPORT	Y	Y	N	N	Y	N
MODDOWN-TIMEHIST	Y	Y	N	N	Y	N
ASSIGNMENT	Y	Y	N	N	Y	N
SERVREC-TRANS	Y	Y	N	N	Y	N
WOSERVICEAD-DRESS	Y	Y	N	N	Y	N
MULTIASSET-LOCCI	Y	Y	N	N	Y	N
COMMLOG	Y	Y	N	N	Y	N
DOCLINKS	Y	Y	N	N	Y	N
MR	Y	Y	N	N	Y	N
MRLINE	Y	Y	N	N	Y	N
WOSAFETYLINK	Y	Y	N	N	Y	Y
PLUSCWODS	Y	Y	N	N	Y	N
PLUSC-WODSINSTR	Y	Y	N	N	Y	N
PLUSC-WODSPOINT	Y	Y	N	N	Y	N
CLASSSTRUCT	Y	Y	N	N	Y	N

4.2.5.8. Create INNO_INVBAL

Create the INNO_INVBAL object structure (Delta INVBALANCES).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_INVBAL
DESCRIPTION	Delta INVBALANCES
Consumed By	INTEGRATION
Authorization Name	INNO_INVBAL
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	Exclude Attributes Mode
INVBALANCES		INVBALANCES		1	N
ITEM	INVBALANCES	INVBALANCES/ITEM	ITEM	1	N
INVENTORY	INVBALANCES	INVBALANCES/INVENTORY	INVENTORY	2	N

Figure 4-13 INNO_INVBAL object structure with its source objects

Object Structure

Object Structure

INNO_INVBAL Delta INVBALANCES

★ Consumed By

INTEGRATION Integration Application

Authorization Name

MXAPIINVBAL

Outbound Definition Class

Inbound Processing Class

Search Attributes

Data restriction condition

Module Name

Auto Paging Threshold

-1

Primary API? ☐

User Defined? ☒

Self Reference? ☐

Configurable? ☒

Publish Complete Hierarchy? ☐

Alias Conflict? ☐

Support Flat Structure? ☐

Load Query From All Apps? ☒

Internal? ☐

Source Objects for INNO_INVBAL (1 - 3 of 3)

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?
INVBALANCES		INVBALANCES		1	☒

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
INVBALANCES	N	N	N	N	N	N
ITEM	Y	Y	N	N	Y	N
INVENTORY	Y	Y	N	N	Y	N

5. From the **Select Action** menu, choose **Configure Object Structure Security**, and select the **Use Object Structure for Authorization Name** check box.

Figure 4-14 Configure Object Structure Security for INNO_INVBAL

Configure Object Structure Security

Object Structure

INNO_INVBAL

Delta INVBALANCES

Use Object Structure for Authorization Name?

☒

Authorization Name

MXAPIINVBAL

Inventory Balance Definition

Add/Modify Signature Options

(1 - 7 of 7)

Select Service Option

	Option	Description	Prerequisite	Visible?	
▼	BMXVIEWMANAGEWI	Sigoption to control the visibility of SQL clause	READ	☒	
▼	DELETE	Delete MXAPIINVBAL	SAVE	☒	
▼	ENABLECOLUMN	Sigoption to control the visibility of column.	READ	☒	
▼	ENABLERECONCILEE	Sigoption to control the visibility of reconcile I	READ	☒	
▼	INSERT	Insert MXAPIINVBAL	SAVE	☒	
▼	READ	Read MXAPIINVBAL		☒	
▼	SAVE	Save MXAPIINVBAL		☒	

1 - 7 of 7

4.2.5.9. Create INNO_LABTRANS

Create the INNO_LABTRANS object structure (Custom OS to send the data Labor trans data).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_LABTRANS
DESCRIPTION	Custom OS to send the data Labor trans data
Consumed By	INTEGRATION
Authorization Name	INNO_LABTRANS
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

Figure 4-15 INNO_LABTRANS object structure definition

The screenshot shows the 'Object Structure' form for INNO_LABTRANS. The form is divided into two main sections: a left sidebar for field definitions and a right panel for configuration options.

Left Sidebar (Field Definitions):

- Object Structure:** INNO_LABTRANS
- DESCRIPTION:** Custom OS to send the data Labor trans data
- Consumed By:** INTEGRATION (with a search icon)
- Authorization Name:** INNO_LABTRANS
- Outbound Definition Class:** (empty field)
- Inbound Processing Class:** (empty field)
- Search Attributes:** (empty field)
- Data restriction condition:** (empty field with a search icon)
- Module Name:** (empty field with a search icon)
- Auto Paging Threshold:** -1

Right Panel (Configuration Options):

- Primary API?:** ☐
- Self Reference?:** ☐
- Publish Complete Hierarchy?:** ☐
- Support Flat Structure?:** ☒
- User Defined?:** ☒
- Configurable?:** ☒
- Alias Conflict?:** ☐
- Load Query From All Apps?:** ☒
- Internal?:** ☐

2. Add **LABTRANS** as the source object (object order 1, no parent object or relationship).

Figure 4-16 LABTRANS as the source object of INNO_LABTRANS

Source Objects for INNO_LABTRANS (1 - 1 of 1)

Object	Parent Object	Object Location Path	Relationship	Object Order
LABTRANS		LABTRANS		1

Details

Object

LABTRANS

Parent Object

Object Location Path

LABTRANS

Exclude Attributes Mode?

Relationship

Object Order

1

Alternate Key

License

User Defined?

1 - 1 of 1

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
LABTRANS	N	N	N	N	N	N

5. From the **Select Action** menu, choose **Configure Object Structure Security**, and select the **Use Object Structure for Authorization Name** check box.

Figure 4-17 Configure Object Structure Security for INNO_LABTRANS

Configure Object Structure Security

Object Structure

INNO_LABTRAN

Custom OS to send the data Labor trans data

Authorization Name

INNO_LABTRANS

Custom OS to send the data Labor trans data

Use Object Structure for Authorization Name?

☒

Add/Modify Signature Options (1 - 5 of 5)

Select Service Option

	Option	Description	Prerequisite	Visible?	
▼	BMXVIEWMANAGEWI	Sigoption to control the visibility of SQL clause	READ	☒	
▼	DELETE	Delete INNO_LABTRANS	SAVE	☒	
▼	INSERT	Insert INNO_LABTRANS	SAVE	☒	
▼	READ	Read INNO_LABTRANS		☒	
▼	SAVE	Save INNO_LABTRANS		☒	

1 - 5 of 5

4.2.5.10. Create INNO_LOCATION

Create the INNO_LOCATION object structure (LOCATION API).

1. In the **Object Structures** application, duplicate **MXAPILOCATION** and enter the following values.

Field	Value
Object Structure	INNO_LOCATION
DESCRIPTION	LOCATION API
Consumed By	INTEGRATION
Authorization Name	MXAPILOCATION
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	ASSET
Auto Paging Threshold	-1
Primary API?	1
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined	Exclude Attributes Mode
LOCATIONS		LOCATIONS		1	Y	N
LOCHIERARCHY	LOCATIONS	LOCATIONS/LOCHIERARCHY	LOCHIERARCHY	1	Y	N
LOCSYSTEM	LOCATIONS	LOCATIONS/LOCSYSTEM	LOCSYSTEM	2	Y	N

Figure 4-18 INNO_LOCATION object structure with its source objects

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?	Exclude Attributes Mode?
LOCATIONS		LOCATIONS		1	Y	N
LOCHIERARCHY	LOCATIONS	LOCATIONS/LOCHIERARCHY	LOCHIERARCHY	1	Y	N
LOCSYSTEM	LOCATIONS	LOCATIONS/LOCSYSTEM	LOCSYSTEM	2	Y	N

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
LOCATIONS	N	N	N	N	N	N
LOCHIERARCHY	Y	Y	N	N	Y	N
LOCSYSTEM	Y	Y	N	N	Y	N

4.2.5.11. Create INNO_USER

Create the INNO_USER object structure (Innovapptive User Details).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_USER
DESCRIPTION	Innovapptive User Details
Consumed By	INTEGRATION
Authorization Name	INNO_USER
Outbound Definition Class	com.ibm.tivoli.maximo.oslc.provider.app.MoutApiKey-Process
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

Figure 4-19 INNO_USER object structure definition

Object Structure

Object Structure
INNO_USER

DESCRIPTION
Innovapptive User Details

Consumed By
INTEGRATION

Authorization Name
INNO_USER

Outbound Definition Class
com.ibm.tivoli.maximo.oslc.provider.app.MoutApiKey

Inbound Processing Class

Search Attributes

Data restriction condition

Module Name

Auto Paging Threshold
-1

Primary API? ☐

Self Reference? ☐

Publish Complete Hierarchy? ☐

Support Flat Structure? ☐

User Defined? ☒

Configurable? ☒

Alias Conflict? ☐

Load Query From All Apps? ☐

Internal? ☐

4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
MAXUSER	N	N	N	N	N	N
PERSON	Y	Y	N	N	Y	N
APIKEYTOKEN	Y	Y	N	N	Y	N
EMAIL	Y	Y	N	N	Y	N
GROUPEUSER	Y	Y	N	N	Y	N
APPLICATION-AUTH	Y	Y	N	N	Y	N

5. From the **Select Action** menu, choose **Configure Object Structure Security**, and select the **Use Object Structure for Authorization Name** check box.

4.2.5.12. Create INNO_RELATEDRECORD

Create the INNO_RELATEDRECORD object structure (Related Records definition - Innovapptive).

1. In the **Object Structures** application, create an object structure with the following values.

Field	Value
Object Structure	INNO_RELATEDRECORD
DESCRIPTION	Related Records definition - Innovapptive
Consumed By	INTEGRATION
Authorization Name	INNO_RELATEDRECORD
Outbound Definition Class	
Inbound Processing Class	
Search Attributes	
Data restriction condition	
Module Name	
Auto Paging Threshold	-1
Primary API?	0
Self Reference?	0
Publish Complete Hierarchy	0
Support Flat Structure	0
Load Query From All Apps?	0

2. Add the following source objects.

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined	Exclude Attributes Mode
RELATE-DRECORD		RELATE-DRECORD		1	Y	N

Figure 4-21 INNO_RELATEDRECORD object structure with its source objects

Object Structure

Object Structure

INNO_RELATED

Related Records definition - Innovapptive

Consumed By

INTEGRATION

Integration Application

Authorization Name

INNO_RELATEDRECORD

Outbound Definition Class

Inbound Processing Class

Search Attributes

Data restriction condition

Module Name

Auto Paging Threshold

-1

Primary API?

Self Reference?

Publish Complete Hierarchy?

Support Flat Structure?

User Defined?

Configurable?

Alias Conflict?

Load Query From All Apps?

Internal?

Source Objects for INNO_RELATEDRECORD (1 - 1 of 1)

Object	Parent Object	Object Location Path	Relationship	Object Order	User Defined?	Exclude Attributes Mode?
RELATE-DRECORD		RELATE-DRECORD		1	☒	☐

3. Leave the other settings (included and excluded fields, aliases, and restrictions) at their default values.
4. From the **Select Action** menu, choose **Advanced Configuration**, and set the following values.

Object	Exclude Parent Key Attributes	Delete Auto-generated Data	Do Not Update Primary Key	Skip delete on replace	Propagate Event	Invoke Execute on Non-persistent Object
RELATE-DRECORD	N	N	N	N	N	N

5. From the **Select Action** menu, choose **Configure Object Structure Security**, and select the **Use Object Structure for Authorization Name** check box.

4.2.6. Configure Object Structure Queries

Create the OSLC queries that filter classifications, locations, and assets.

1. In the **Object Structures** application, open **INNO_CLASSIFICATION** , choose **Query Definitions** from the **Select Action** menu, and create the following queries.

Object structure	Query type	Query clause name	Public	QBE clause
INNO_CLASSIFI-CATION	osclause	classificationasset	Y	N
INNO_CLASSIFI-CATION	osclause	classificationshr	Y	N
INNO_CLASSIFI-CATION	osclause	classificationwo	Y	N

classificationasset

```
exists (select 1 from classusewith where classstructureid=classstructure.classstructureid and objectvalue='ASSET')
```

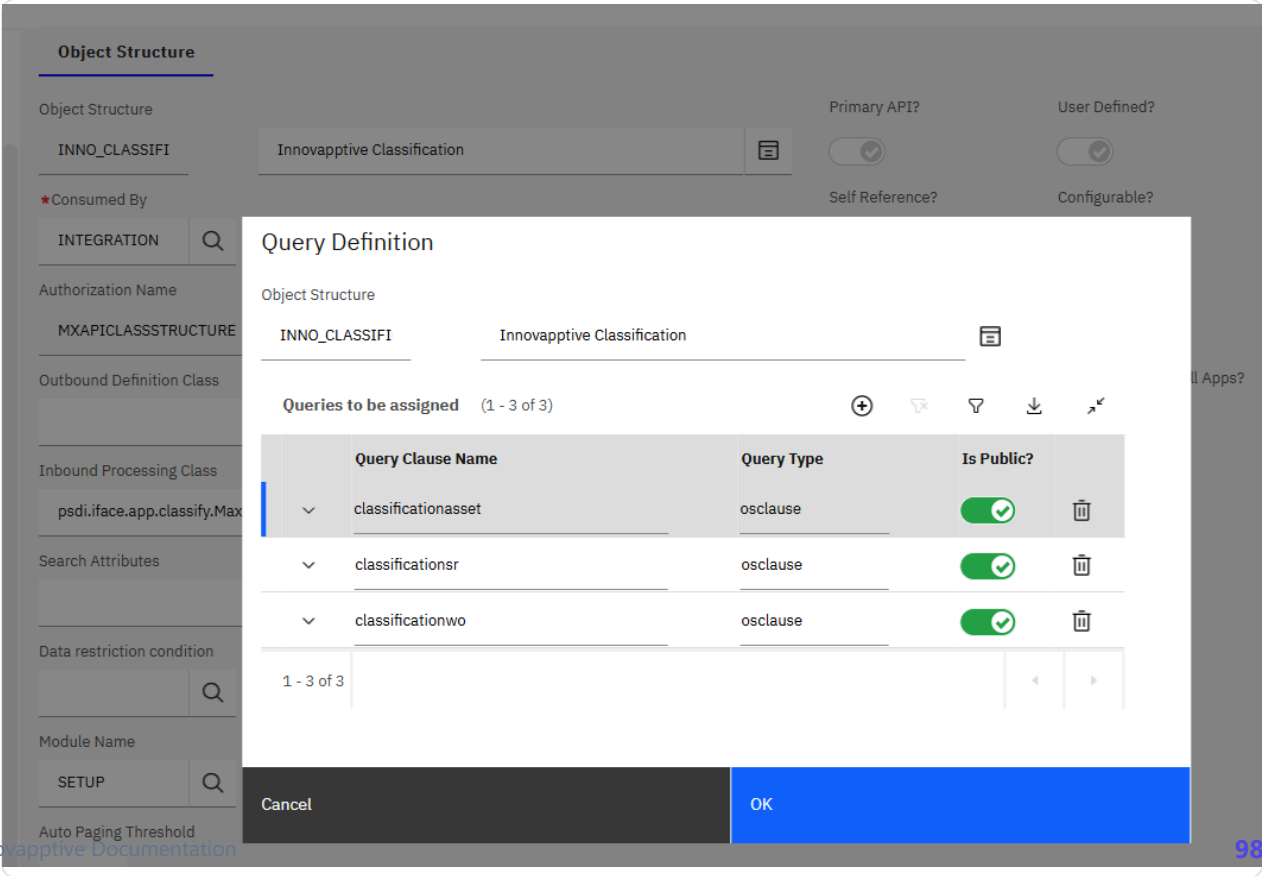
classificationshr

```
exists (select 1 from classusewith where classstructureid=classstructure.classstructureid and objectvalue='SR')
```

classificationwo

```
exists (select 1 from classusewith where classstructureid=classstructure.classstructureid and objectvalue='WORKORDER')
```

Figure 4-22 Query definitions on INNO_CLASSIFICATION



2. Open **INNO_LOCATION** and **MXAPIASSET** and create the location filter queries.

Object structure	Query type	Query clause name	Public	QBE clause
INNO_LOCATION	osclause	INNO_LOC_MAINT_FILTER	Y	N
MXAPIASSET	osclause	INNO_ASSET_MAIN-T_FILTER	Y	N

INNO_LOC_MAINT_FILTER

```
location not like '<location-prefix>%' and location not in ('<location>')
```

INNO_ASSET_MAINT_FILTER

```
location is not null and location not like '<location-prefix>%' and location not in ('<location>')
```

Repeat the **not like** condition for each location prefix to exclude, and list each excluded location in the **not in** list. The asset query also keeps only assets that have a location.

4.2.7. Create Integration Controls

Create the list controls that limit which sites, locations, and attachment owners send data.

INNOSKIPSITE lists the sites that are allowed to send data. **MAINT_LOC_EXCLUDE** and **MAINT_LOC_EXCLUDE1** list the locations and location prefixes to exclude.

In the **Publish Channels** application, choose **Create Integration Controls** from the **Select Action** menu, and create the following list controls.

Type	Integration control	Description	Values
List Control	INNOSKIPDOCLINKS	Skip doclinks trigger other than WO, ASSET and SR	ASSET, SR, WORKORDER, WOACTIVITY
List Control	MAINT_LOC_EXCLUDE	Skip the following exact location records	<location> (one entry per location to exclude)
List Control	MAINT_LOC_EXCLUDE1	Skip the following location like records	<location-prefix>% (one entry per location prefix to exclude)
List Control	INNOSKIPSITE	List of Sites that are allowed to send data - Innovapptive	<site-id> (one entry per site allowed to send data)

Add each value as a separate list entry.

4.2.8. Create the iMaintenance Publish Channels

Create the nine publish channels that send Maximo changes to iMaintenance, with their processing rules.

Each publish channel sends changes from one object structure. Processing rules skip records the integration must not send, such as records changed by the integration user or records from sites that are not allowed.

- [Create the INNO_ASSET Publish Channel \(on page 100\)](#)
- [Create the INNO_ASSIGNMENT Publish Channel \(on page 103\)](#)
- [Create the INNO_ATTACHMENT Publish Channel \(on page 107\)](#)
- [Create the INNO_LABTRANS Publish Channel \(on page 112\)](#)
- [Create the INNO_LOCATION Publish Channel \(on page 115\)](#)
- [Create the INNO_WO Publish Channel \(on page 118\)](#)
- [Create the INNO_WOACT Publish Channel \(on page 122\)](#)
- [Create the INNV_WORKLOG Publish Channel \(on page 126\)](#)
- [Create the INNO_RELATEDRECORD Publish Channel \(on page 130\)](#)

4.2.8.1. Create the INNO_ASSET Publish Channel

Create the INNO_ASSET publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_ASSET
Description	Custom PS to send the data to iMaintenance
Operation	Publish
Object Structure	INNO_ASSET
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbo's	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-23 INNO_ASSET publish channel

Publish Channel

Publish Channel

INNO_ASSET

Custom PS to send the data to iMaintenance

Operation

Publish

Object Structure

INNO_ASSET

Processing Class

Event Filter Class

Adapter

MAXIMO

Interface Table

XSL Map

User Exit Class

Template Name

User Defined?

Retain Mbo's?

Enable Listener?

Skip Different Object Types?

Publish JSON?

Object Structure Sub-Records (1 - 5 of 6)

Object	Object Location Path	Processing Order
ASSET	ASSET	1
ASSETMETER	ASSET/ASSETMETER	2
ASSETSPEC	ASSET/ASSETSPEC	3
DOCLINKS	ASSET/DOCLINKS	4
DOWNTIMEREPORT	ASSET/DOWNTIMEREPORT	5

1 - 5 of 6

Processing Rules for Sub-Record (0 - 0 of 0)

Rule	Description	Sequence	Action	Enabled?	User Defined?
------	-------------	----------	--------	----------	---------------

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP1** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	
SITEID	EQUALS	ALWAYS		<site-id>

4.2.8.2. Create the INNO_ASSIGNMENT Publish Channel

Create the INNO_ASSIGNMENT publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_ASSIGNMENT
Description	Custom Publish Channel to send out WO's Assignment
Operation	Publish
Object Structure	INNO_ASSIGNMENT
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbos	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-24 INNO_ASSIGNMENT publish channel and its processing rules

Publish Channel

Publish Channel: INNO_ASSIGNMENT
Description: Custom Publish Channel to send out WO's Assignment

*Operation: Publish

Object Structure: INNO_ASSIGNMENT

Processing Class:

Event Filter Class:

*Adapter: MAXIMO

Interface Table:

XSL Map:

User Exit Class:

Template Name:

User Defined? ☒

Retain Mbo's? ☒

Enable Listener? ☐

Skip Different Object Types? ☐

Publish JSON? ☒

Object Structure Sub-Records (1 - 1 of 1)

Object	Object Location Path	Processing Order
ASSIGNMENT	ASSIGNMENT	1

1 - 1 of 1

Processing Rules for Sub-Record ASSIGNMENT (1 - 2 of 2)

Rule	Description	Sequence	Action	Enabled?	User Defined?
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	☒	☒
INNO_3M_SKIP2	Skip Transactions of some business lines -3M	2	SKIP	☒	☒

1 - 2 of 2

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP1** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	Y	N
20	Y	Y	N	N
30	Y	Y	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Set Evaluations for Condition 10

Object	Object Relationship	Evaluate Type
LOCATIONS	INNO_LOCATION	NOTEXISTS

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Field Evaluations for Condition 20

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
LOCATIONS	INNO_LOCATION	LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	

XML Field Evaluations for Condition 30

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Field Evaluations for Condition 30

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
LOCATIONS	INNO_LOCATION	LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	

4.2.8.3. Create the INNO_ATTACHMENT Publish Channel

Create the INNO_ATTACHMENT publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_ATTACHMENT
Description	Innovapptive Custom PS to Publish Channel to push out Attachment
Operation	Publish
Object Structure	INNO_ATTACHMENT
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbos	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-25 INNO_ATTACHMENT publish channel and its processing rules

Publish Channel

Publish Channel

INNO_ATTACHMENT

Innovapptive Custom PS to Publish Channel to push out Attachm

*Adapter

MAXIMO

User Defined?

☒

*Operation

Publish

Interface Table

Object Structure

INNO_ATTACHMENT

XSL Map

Processing Class

User Exit Class

Event Filter Class

Template Name

Retain Mbo's?

☒

Enable Listener?

☒

Skip Different Object Types?

☐

Publish JSON?

☒

Object Structure Sub-Records (1 - 1 of 1)

Object	Object Location Path	Processing Order
DOCLINKS	DOCLINKS	1

Processing Rules for Sub-Record DOCLINKS (1 - 3 of 3)

Rule	Description	Sequence	Action	Enabled?	User Defined?
INNO_SKIP2	Skip outbound integration events for transacti	1	SKIP	☒	☐
INNO_SET1	Set the Document data as NULL - Innovappti	2	SET	☒	☐
INNO_3M_SKIP2	Skip Transactions of some business lines -3M	3	SKIP	☒	☐

2. Associate the integration controls `INNOSKIPDOCLINKS` , `INNOSKIPSITE` , `MAINT_LOC_EXCLUDE` , `MAINT_LOC_EXCLUDE1` with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.

4. Create the processing rule **INNO_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP2	Skip outbound integration events for transactions inbound from Maximo Mobile. - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N
30	N	Y	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
CREATEBY	EQUALS	ALWAYS		INNOVAPPTIVE

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
OWNERTABLE	NOTEQUALS	ALWAYS	INNOSKIPDOCLINKS	

Object Field Evaluations for Condition 30

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
SITE	INNO_SITE	SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_SET1**.

Rule	Description	Sequence	Action	Enabled
INNO_SET1	Set the Document data as NULL - Innovapptive	2	SET	Y

Set Sub-Record Fields

Field	Value
DOCUMENTDATA	US

6. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	3	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	N	Y	Y	N
20	N	Y	N	N
30	N	Y	N	N

Object Field Evaluations for Condition 10

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
SITE	INNO_SITE	SITEID	EQUALS	ALWAYS		<site-id>

Object Set Evaluations for Condition 10

Object	Object Relationship	Evaluate Type
LOCATIONS	INNO_LOCATION	NOTEXISTS

Object Field Evaluations for Condition 20

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
SITE	INNO_SITE	SITEID	EQUALS	ALWAYS		<site-id>
LOCATIONS	INNO_LOCATION	LOCATION	EQUALS	ALWAYS	MAINT_LOC_-EXCLUDE	

Object Field Evaluations for Condition 30

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
SITE	INNO_SITE	SITEID	EQUALS	ALWAYS		<site-id>
LOCATIONS	INNO_LOCATION	LOCATION	LIKE	ALWAYS	MAINT_LOC_-EXCLUDE1	

4.2.8.4. Create the INNO_LABTRANS Publish Channel

Create the INNO_LABTRANS publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_LABTRANS
Description	Custom PS to send the Labtrans data to iMaintenance
Operation	Publish
Object Structure	INNO_LABTRANS
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbo's	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON?	1

Figure 4-26 INNO_LABTRANS publish channel and its processing rules

Publish Channel

Publish Channel: INNO_LABTRAN

Description: Custom PS to send the Labtrans data to iMaintenance

*Operation: Publish

Object Structure: INNO_LABTRAN

Processing Class:

Event Filter Class:

*Adapter: MAXIMO

Interface Table:

XSL Map:

User Exit Class:

Template Name:

User Defined?: ☒

Retain Mbo's?: ☒

Enable Listener?: ☒

Skip Different Object Types?: ☐

Publish JSON?: ☒

Object Structure Sub-Records (1 - 1 of 1)

Object	Object Location Path	Processing Order
LABTRANS	LABTRANS	1

1 - 1 of 1

Processing Rules for Sub-Record LABTRANS (1 - 2 of 2)

Rule	Description	Sequence	Action	Enabled?	User Defined?
INNO_SKIP2	Skip outbound integration events for transacti	1	SKIP	☒	☒
INNO_3M_SKIP2	Skip Transactions of some business lines -3M	2	SKIP	☒	☒

1 - 2 of 2

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP2** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP2	Skip outbound in- tegration events for transactions in- bound from Maximo Mobile. - Innovapp- tive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
ENTERBY	EQUALS	ALWAYS		INNOVAPPTIVE

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS		INNOSKIPSITE

5. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	

4.2.8.5. Create the INNO_LOCATION Publish Channel

Create the INNO_LOCATION publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_LOCATION
Description	Custom PS to update
Operation	Publish
Object Structure	INNO_LOCATION
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbo's	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-27 INNO_LOCATION publish channel

Publish Channel

Publish Channel: INNO_LOCATION | Custom PS to update

*Operation: Publish

Object Structure: INNO_LOCATION

Processing Class:

Event Filter Class:

*Adapter: MAXIMO

Interface Table:

XSL Map:

User Exit Class:

Template Name:

User Defined?: ☒

Retain Mbo's?: ☒

Enable Listener?: ☒

Skip Different Object Types?: ☐

Publish JSON?: ☒

Object Structure Sub-Records (1 - 3 of 3)

Object	Object Location Path	Processing Order
LOCATIONS	LOCATIONS	1
LOCHIERARCHY	LOCATIONS/LOCHIERARCHY	2
LOCSYSTEM	LOCATIONS/LOCSYSTEM	3

1 - 3 of 3

Processing Rules for Sub-Record (0 - 0 of 0)

Rule	Description	Sequence	Action	Enabled?	User Defined?
------	-------------	----------	--------	----------	---------------

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP1** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	
SITEID	EQUALS	ALWAYS		<site-id>

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	
SITEID	EQUALS	ALWAYS		<site-id>

4.2.8.6. Create the INNO_WO Publish Channel

Create the INNO_WO publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_WO
Description	Custom Publish channel to push out the WO change
Operation	Publish
Object Structure	INNO_GETWO
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbos	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-28 INNO_WO publish channel

Publish Channel

Publish Channel: INNO_WO, Custom Pulish channel to push out the WO change

*Adapter: MAXIMO

*Operation: Publish

Object Structure: INNO_GETWO

Processing Class:

Event Filter Class:

Interface Table:

XSL Map:

User Exit Class:

Template Name:

User Defined? ☒

Retain Mbo's? ☒

Enable Listener? ☒

Skip Different Object Types ☐

Publish JSON? ☒

Object Structure Sub-Records (1 - 5 of 30)

Object	Object Location Path	Processing Order
WORKORDER	WORKORDER	1
WPLABOR	WORKORDER/WPLABOR	2
WPMATERIAL	WORKORDER/WPMATERIAL	3
WPTOOL	WORKORDER/WPTOOL	4
WPSERVICE	WORKORDER/WPSERVICE	5

1 - 5 of 30

Processing Rules for Sub-Record WORKORDER (1 - 4 of 4)

Actions

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP2** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP2	Skip outbound integration events for transactions inbound from Maximo Mobile. - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
CHANGEBY	EQUALS	ALWAYS		INNOVAPPTIVE

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_SKIP3**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP3	Skip the Transactions when WOCLASS != "WORKORDER" - Innovapptive.	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
WOCLASS	NOTEQUALS	ALWAYS		WORKORDER

6. Create the processing rule **INNO_SKIP4**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP4	Skip all the Transactions when WOJO1 is equal to mobile - Innovapptive	3	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
WOJO1	EQUALS	ALWAYS		mobile

7. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	4	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	

4.2.8.7. Create the INNO_WOACT Publish Channel

Create the INNO_WOACT publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_WOACT
Description	Custom Publish channel to push out the WO change specially Activities
Operation	Publish
Object Structure	INNO_GETWO
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbo's	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-29 INNO_WOACT publish channel

The screenshot shows the 'Publish Channel' configuration page for 'INNO_WOACT'. The interface is divided into several sections:

- Publish Channel:** Shows the channel name 'INNO_WOACT' and its description 'Custom Pulish channel to push out the WO change specially Activ'.
- Operation:** A dropdown menu set to 'Publish'.
- Object Structure:** A dropdown menu set to 'INNO_GETWO'.
- Processing Class:** An empty text field.
- Event Filter Class:** An empty text field.
- Adapter:** A dropdown menu set to 'MAXIMO'.
- Interface Table:** An empty text field.
- XSL Map:** An empty text field.
- User Exit Class:** An empty text field.
- Template Name:** An empty text field.
- User Defined?:** A toggle switch that is turned on.
- Retain Mbo's?:** A toggle switch that is turned on.
- Enable Listener?:** A toggle switch that is turned on.
- Skip Different Object Types?:** A toggle switch that is turned off.
- Publish JSON?:** A toggle switch that is turned on.

Below the configuration fields, there is a table titled 'Object Structure Sub-Records (1 - 5 of 30)'. The table has three columns: 'Object', 'Object Location Path', and 'Processing Order'.

Object	Object Location Path	Processing Order
WORKORDER	WORKORDER	1
WPLABOR	WORKORDER/WPLABOR	2
WPMATERIAL	WORKORDER/WPMATERIAL	3
WPTOOL	WORKORDER/WPTOOL	4
WPSERVICE	WORKORDER/WPSERVICE	5

At the bottom of the page, there is a section titled 'Processing Rules for Sub-Record WORKORDER (1 - 3 of 3)' with a table and an 'Actions' button.

2. Associate the integration controls `INNOSKIPSITE` , `MAINT_LOC_EXCLUDE` , `MAINT_LOC_EXCLUDE1` with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.

4. Create the processing rule **INNO_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP2	Skip outbound integration events for transactions inbound from Maximo Mobile. - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
CHANGEBY	EQUALS	ALWAYS		INNOVAPPTIVE

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_SKIP5**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP5	Skip the Transactions when WOCLASS == "WORKORDER" - Innovapptive.	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
WOCLASS	EQUALS	ALWAYS		WORKORDER

6. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	3	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	EQUALS	ALWAYS	MAINT_LOC_EX-CLUE	

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>
LOCATION	ISNOTNULL	ALWAYS		
LOCATION	LIKE	ALWAYS	MAINT_LOC_EX-CLUE1	

4.2.8.8. Create the INNV_WORKLOG Publish Channel

Create the INNV_WORKLOG publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNV_WORKLOG
Description	Custom PC to send Work logs update to iMaintenance
Operation	Publish
Object Structure	INNV_WORKLOG
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbo's	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON?	1

Figure 4-30 INNV_WORKLOG publish channel and its processing rules

Publish Channel

Publish Channel: INNV_WORKLOG
Description: Custom PC to send Work logs update to iMaintenance
Operation: Publish
Object Structure: INNV_WORKLOG
Processing Class:
Event Filter Class:
Adapter: MAXIMO
Interface Table:
XSL Map:
User Exit Class:
Template Name:
User Defined?:
Retain Mbo's?:
Enable Listener?:
Skip Different Object Types:
Publish JSON?:

Object Structure Sub-Records (1 - 1 of 1)

Object	Object Location Path	Processing Order
WORKLOG	WORKLOG	1

1 - 1 of 1

Processing Rules for Sub-Record WORKLOG (1 - 2 of 2)

Rule	Description	Sequence	Action	Enabled?	User Defined?
INNO_SKIP2	Skip outbound integration events for transacti	1	SKIP	☒	☒
INNO_3M_SKIP2	Skip Transactions of some business lines -3M	2	SKIP	☒	☒

1 - 2 of 2

2. Associate the integration controls **INNOSKIPSITE** , **MAINT_LOC_EXCLUDE** , **MAINT_LOC_EXCLUDE1** with the publish channel.
3. For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
4. Create the processing rule **INNO_SKIP2** .

Rule	Description	Sequence	Action	Enabled
INNO_SKIP2	Skip outbound in- tegration events for transactions in- bound from Maximo Mobile. - Innovapp- tive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
CREATEBY	EQUALS	ALWAYS		INNOVAPPTIVE

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

5. Create the processing rule **INNO_3M_SKIP2**.

Rule	Description	Sequence	Action	Enabled
INNO_3M_SKIP2	Skip Transactions of some business lines -3M – Innovapptive	2	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	Y	N
20	Y	Y	N	N
30	Y	Y	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Set Evaluations for Condition 10

Object	Object Relationship	Evaluate Type
LOCATIONS	INNO_LOCATION	NOTEXISTS

XML Field Evaluations for Condition 20

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Field Evaluations for Condition 30

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
LOCATIONS	INNO_LOCATION	LOCATION	EQUALS	ALWAYS	MAINT_LOC_EXCLUDE	

XML Field Evaluations for Condition 30

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	EQUALS	ALWAYS		<site-id>

Object Field Evaluations for Condition 30

Object	Object Relationship	Field	Evaluation Type	Evaluation When	Integration Control	Value
LOCATIONS	INNO_LOCATION	LOCATION	LIKE	ALWAYS	MAINT_LOC_EXCLUDE1	

4.2.8.9. Create the INNO_RELATEDRECORD Publish Channel

Create the INNO_RELATEDRECORD publish channel and its processing rules for iMaintenance.

Values in angle brackets are deployment inputs.

1. In the **Publish Channels** application, create a publish channel with the following values.

Field	Value
Publish Channel	INNO_RELATEDRECORD
Description	Custom PC to send Related Record update to iMaintenance - Innovapptive
Operation	Publish
Object Structure	INNO_RELATEDRECORD
Processing Class	
Event Filter Class	
Adapter	Maximo
Interface Table	
XSL Map	
User Exit Class	
Template Name	
Retain Mbos	1
Enable Listener	1
Skip Different Object Types	0
Publish JSON	1

Figure 4-31 INNO_RELATEDRECORD publish channel

The screenshot displays the 'Publish Channel' configuration page for 'INNO_RELATEDRECORD'. The configuration is as follows:

- Publish Channel:** INNO_RELATEDRECORD
- Description:** Custom PC to send Related Record update to iMaintenance - Inn
- Operation:** Publish
- Object Structure:** INNO_RELATEDRECORD
- Processing Class:** (Empty)
- Event Filter Class:** (Empty)
- Adapter:** MAXIMO
- Interface Table:** (Empty)
- XSL Map:** (Empty)
- User Exit Class:** (Empty)
- Template Name:** (Empty)
- User Defined?:** ☒
- Retain Mbo's?:** ☒
- Enable Listener?:** ☒
- Skip Different Object Types?:** ☐
- Publish JSON?:** ☒

Object Structure Sub-Records (1 - 1 of 1)

Object	Object Location Path	Processing Order
RELATEDRECORD	RELATEDRECORD	1

Processing Rules for Sub-Record RELATEDRECORD (1 - 1 of 1)

Rule	Description	Sequence	Action	Enabled?	User Defined?
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	☒	☒

- Associate the integration controls **INNOSKIPSITE** with the publish channel.
- For every processing rule below, select **Apply on Primary Object Insert**, **Apply on Primary Object Update**, and **Apply on Primary Object Delete**. Configure each rule on the primary object of the object structure.
- Create the processing rule **INNO_SKIP1**.

Rule	Description	Sequence	Action	Enabled
INNO_SKIP1	Rule to skip site level records - Innovapptive	1	SKIP	Y

Add/Modified Conditions

Condition	XML Field Evaluation	Object Field Evaluation	Object Set Evaluation	Control Evaluation
10	Y	N	N	N
20	Y	N	N	N

XML Field Evaluations for Condition 10

Field	Evaluation Type	Evaluate When	Integration Control	Value
SITEID	NOTEQUALS	ALWAYS	INNOSKIPSITE	

4.2.9. Configure Maximo Kafka Endpoints

Route Maximo integration messages to the Kafka topics supplied for the deployment.

Before you begin

The Innovapptive infrastructure team has supplied the Kafka bootstrap server and port, the user name, and the password for this deployment.

Replace **<bootstrap-server>:<port>**, **<kafka-username>**, and **<kafka-password>** with the supplied values. Enter the password as an encrypted value. Keep the topic names as shown.

1. Open the **End Points** application.
2. Create the endpoint **INNO_MASTER**.

End Point	Description	Handler	Consumed By
INNO_MASTER	Custom Kafka topic for WO transactions	KAFKA	INTEGRATION

Property	Value	Encrypted Value	Allow Override
BOOTSTRAPSERVERS	<bootstrap-server>:<port>		N
ISTEXT			N
KERBEROS_SERVICE			N
KEYTABFILE			N
KEYTABPRINCIPAL			N
PASSWORD		<kafka-password>	N
SASL_MECHANISM	SCRAM-SHA-512		N
SECURITY_PROTOCOL	SASL_SSL		N
TOPIC	MasterDataTopic		N
USERNAME	<kafka-username>		N

3. Create the endpoint **INNO_WO**.

End Point	Description	Handler	Consumed By
INNO_WO	Custom Kafka topic for WO transactions	KAFKA	INTEGRATION

Property	Value	Encrypted Value	Allow Override
BOOTSTRAPSERVERS	<bootstrap-server>:<port>		N
ISTEXT			N
KERBEROS_SERVICE			N
KEYTABFILE			N
KEYTABPRINCIPAL			N
PASSWORD		<kafka-password>	N
SASL_MECHANISM	SCRAM-SHA-512		N
SECURITY_PROTOCOL	SASL_SSL		N
TOPIC	WorkOrderTopic		N
USERNAME	<kafka-username>		N

4. Create the endpoint **INNO_WOENTITIES**.

End Point	Description	Handler	Consumed By
INNO_WOENTITIES	Custom Kafka topic for WO Entities transactions	KAFKA	INTEGRATION

Property	Value	Encrypted Value	Allow Override
BOOTSTRAPSERVERS	<bootstrap-server>:<port>		N
ISTEXT			N
KERBEROS_SERVICE			N
KEYTABFILE			N
KEYTABPRINCIPAL			N
PASSWORD		<kafka-password>	N
SASL_MECHANISM	SCRAM-SHA-512		N
SECURITY_PROTOCOL	SASL_SSL		N
TOPIC	WorkOrderEntitiesTopic		N
USERNAME	<kafka-username>		N

4.2.10. Configure the Kafka Message Provider

Import the Kafka certificate into MAS and create the KAFKA message provider in Maximo.

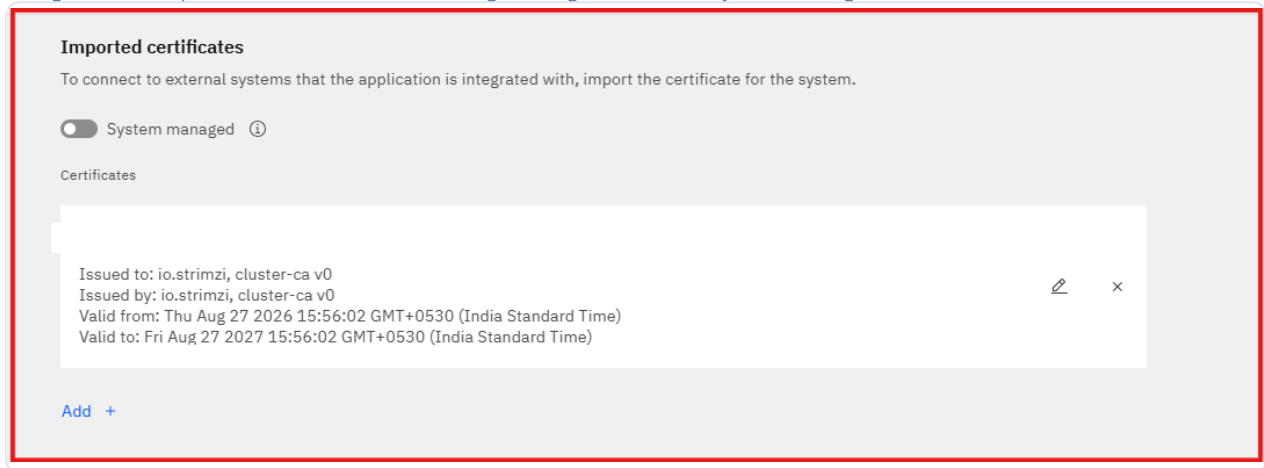
Before you begin

The Innovapptive infrastructure team has supplied the certificate file, the Kafka connection URL, and the topic details for this environment.

If iMaintenance and mRounds use the same MAS environment, set up the certificate and provider once.

1. In MAS Suite administration, open the Manage configuration. Under **Imported certificates**, turn off **System managed**, click **Add**, and add the Kafka certificate.

Figure 4-32 Imported certificates in the Manage configuration, with System managed turned off (certificate alias masked)



2. Apply the changes and activate the configuration.
3. In the OpenShift console, monitor the recreation of the `manage-all` pod to confirm that the certificate is imported.

4. In the **External Systems** application, choose **Add/Modify Message Providers** from the **Select Action** menu, and create the following provider.

Name	Provider Type	Provider Type Description	Active
KAFKA	KAFKA	Kafka Message Provider	Y

Property	Value	Encrypted Value
KEYTABPRINCIPAL		
KEYTABFILE		
KERBEROS_SERVICE		
SASL_MECHANISM	SCRAM-SHA-512	
SECURITY_PROTOCOL	SASL_SSL	
BOOTSTRAPSERVERS	<bootstrap-server>:<port>	
PASSWORD		<kafka-password>
USERNAME	<kafka-username>	

Figure 4-33 KAFKA message provider, properties 1 to 5

Message Hub Provider

Message Hub Provider (1 - 1 of 1)

</

4.2.11. Create the Maximo Queues

Create the inbound and outbound queues that use the KAFKA message provider.

CQIN and **CQINERR** are inbound continuous queues. **CQOUT** is an outbound continuous queue, and **SQOUT** is an outbound sequential queue. These are Maximo queues, not Kafka topics.

In the **External Systems** application, choose **Add/Modify Queues** from the **Select Action** menu, and create the following queues.

Queue JNDI Name	Sequential	Inbound	User Defined	Text	Maximum Try Count	Message Provider
CQIN	N	Y	Y	N	5	KAFKA
CQINERR	N	Y	Y	N	5	KAFKA
CQOUT	N	N	Y	N	5	KAFKA
SQOUT	Y	N	Y	N	5	KAFKA

Figure 4-35 Kafka queues: partial view (first 5 of 13). The table above lists them all.

Add/Modify Queues

Queues (1 - 5 of 13)

	Queue JNDI Name		Sequential ?	Inbound?	User Defined?	Text?	Maximum Try Count	Message Provider		
✓	CQIN	Q	☐	☒	☒		5	KAFKA	Q	🗑
✓	CQINERR	Q	☐	☒	☒		5	KAFKA	Q	🗑
✓	CQOUT	Q	☐	☐	☒		5	KAFKA	Q	🗑
✓	SQOUT	Q	☒	☐	☒		5	KAFKA	Q	🗑

4.2.12. Create the Maximo Automation Scripts

Create the integration scripts and attach their Maximo launch points.

Enter the script source exactly as shown.

1. Open the **Automation Scripts** application.
2. Create the script **INNO_ASSIGNMENT** with the following source.

Field	Value
Script name	INNO_ASSIGNMENT
Description	Custom Script to copy the workorderid from wo to as- signment
Script language	Jython
Log level	ERROR
Active	Yes
Variables	None

```
wo = mbo.getOwner()  
mbo.setValue("im workorderid", wo.getDouble("workorderid"))
```

3. Add the launch point for `INNO_ASSIGNMENT`.

Field	Value
Launch point name	INNO_ASSIGNMENT
Description	Custom Script to copy the workorderid from wo to assignment
Launch point type	OBJECT
Object	ASSIGNMENT
Active	Yes
Events	Save
Save	Add
Before Save	Yes

Figure 4-36 Launch point for INNO_ASSIGNMENT

[illegible]

4. Create the script **INNO_CHANGEDATE** with the following source.

Field	Value
Script name	INNO_CHANGEDATE
Description	Populate INNO_CHANGEDATE
Script language	Jython
Log level	ERROR
Active	Yes
Variables	None

```
from psdi.server import MXServer
from psdi.mbo import MboConstants

maximo = MXServer.getMXServer()
mbo.setFlag(MboConstants.READONLY, False)
mbo.setValue("INNO_CHANGEDATE",maximo.getDate(),MboConstants.NOACCESSCHECK)
```

5. Add the following object launch points for **INNO_CHANGEDATE** . For each one, set **Active** to Yes, **Events** to Save, **Save** to Add and Update, and **Before Save** to Yes.

Launch point name	Description	Object
ITEM	ITEM	ITEM
PERSON	PERSON	PERSON
LABOR	LABOR	LABOR
INVENTORY	INVENTORY	INVENTORY
WPMATERIAL	WPMATERIAL	WPMATERIAL
INVBALANCES	INVBALANCES	INVBALANCES

Figure 4-37 One of the six INNO_CHANGEDATE launch points (ITEM). The table lists all six.

Object Launch Point

Launch Point

ITEM

ITEM

Active?

ITEM

Q

Object Event Condition

Events

Initialize Value

Validate Application

Allow Object Creation

Allow Object Deletion

Save

Before Save

After Save

After Commit

Add?

Update?

Delete?

Save

Launch Point Variables

(0 - 0 of 0)

Variable	Variable Type	Binding Type	Binding Value	Overridden?
----------	---------------	--------------	---------------	-------------

Innovapptive Documentation

141

6. Create the script `INNO_WO` with the following source.

Field	Value
Script name	INNO_WO
Description	Populate Custom Auto Script to update WOJO1 in Work Order
Script language	Jython
Log level	ERROR
Active	Yes
Variables	None

```
if interactive == True:
    if mbo.getString("WOJO1") == "mobile":
        mbo.setValueNull("WOJO1")
```

7. Add the launch point for `INNO_WO`.

Field	Value
Launch point name	INNO_WO
Description	Custom LP to update WOJO1 in Work Order
Launch point type	OBJECT
Object	WORKORDER
Active	Yes
Events	Save
Save	Add, Update
Before Save	Yes

Figure 4-38 Launch point for INNO_WO

Object Launch Point

Launch Point

INNO_WO

Custom LP to update WOJO1 in Work Order

Active?

Object

WORKORDER

Object Event Condition

Events

Initialize Value

Validate Application

Allow Object Creation

Allow Object Deletion

Save

Before Save

After Save

After Commit

Add?

Update?

Delete?

Save

Launch Point Variables

(0 - 0 of 0)

Variable

Variable Type

Binding Type

Binding Value

Overridden?

8. Create the script `INNO_WOCHANGE` with the following source.

Field	Value
Script name	INNO_WOCHANGE
Description	Custom Script to update the change date
Script language	Jython
Log level	ERROR
Active	Yes
Variables	None

```
from psdi.server import MXServer
from psdi.mbo import MboConstants

maximo = MXServer.getMXServer()
WOSet = mbo.getMboSet("WORKORDER", "WORKORDER", "wonum =:recordkey")
if not WOSet.isEmpty():
    WO = WOSet.moveFirst()
    WO.setValu("CHANGEDATE", maximo.nowDate())
```

9. Add the launch point for **INNO_WOCHANGE** .

Field	Value
Launch point name	INNO_WOCHANGE
Description	Custom LP to change the date to worklog
Launch point type	OBJECT
Object	WORKLOG
Active	Yes
Events	Save
Save	Add
Before Save	Yes

Figure 4-39 Launch point for INNO_WOCHANGE

Object Launch Point

Launch Point

INNO_WOCHAN

Custom LP to change the date to worklog

Active?

Object

WORKLOG

Object Event Condition

Events

Initialize Value

Validate Application

Allow Object Creation

Allow Object Deletion

Save

Add?

Update?

Delete?

Before Save

After Save

After Commit

Launch Point Variables

(0 - 0 of 0)

Variable	Variable Type	Binding Type	Binding Value	Overridden?
----------	---------------	--------------	---------------	-------------

Innovapptive Documentation

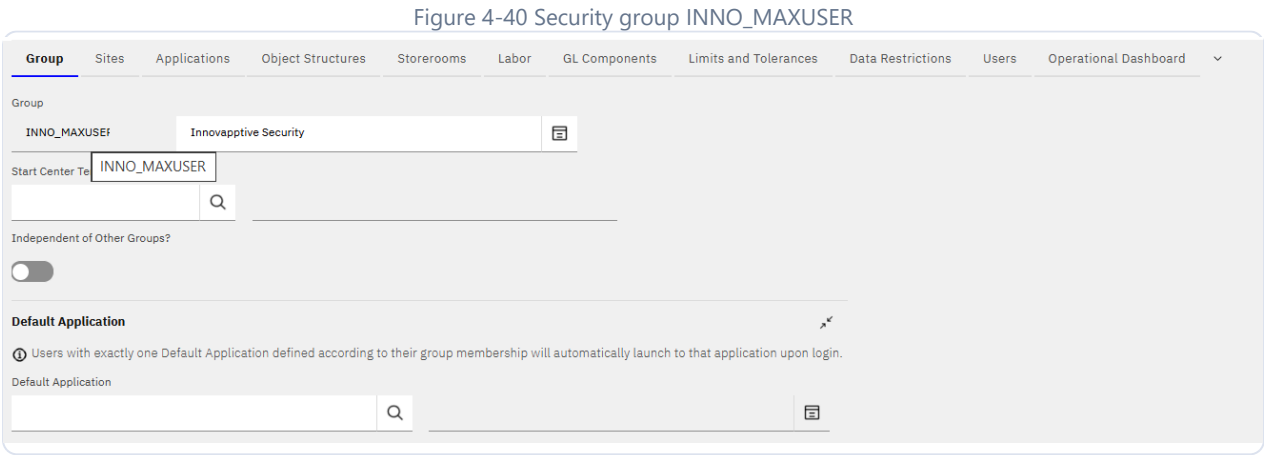
145

4.2.13. Configure Maximo Roles and Authorizations

Create the INNO_MAXUSER security group, give it site access and object structure permissions, and add the integration user.

This group gives the integration user its runtime access in Maximo. The administrator who configures Maximo needs separate administrative access, such as **MAXADMIN**.

1. In the **Security Groups** application, create the group **INNO_MAXUSER** .



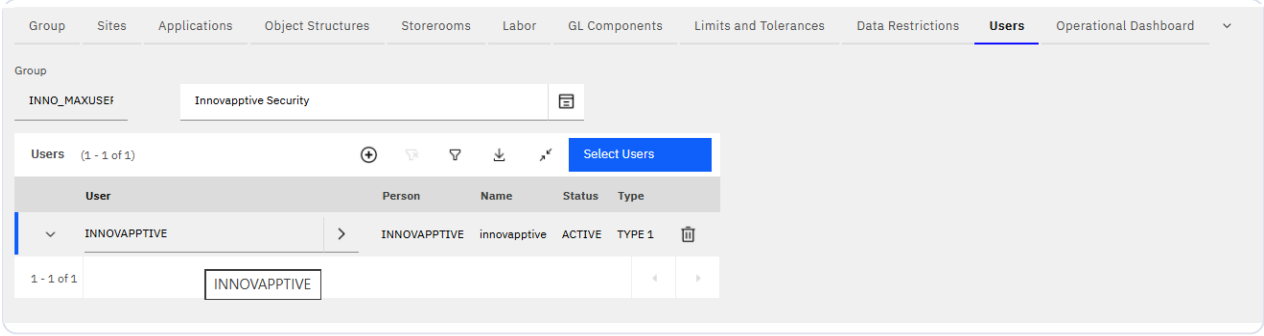
2. On the **Sites** tab, add **<site-id>** .

3. On the **Object Structures** tab, grant the following access.

OS name	Authorization name	Access Grants
MXAPIORGANIZATION	MXAPIORGANIZATION	Read
MXAPIASSET	MXAPIASSET	Read
INNO_LOCATION	MXAPILOCATION	Read
MXAPICRAFT	MXAPICRAFT	Read
MXAPIMEASUREUNIT	MXAPIMEASUREUNIT	Read
MXAPIPERSON	MXAPIPERSON	Read
MXAPILABOR	MXAPILABOR	Read
MXAPIPERSONGROUP	MXAPIPERSONGROUP	Read
MXAPISYNONYMDOMAIN	MXAPISYNONYMDOMAIN	Read
MXAPIWORKTYPE	MXAPIWORKTYPE	Read
MXAPIITEM	MXAPIITEM	Read
INNO_INVBAL	MXAPIINVBAL	Read
MXAPIASSETSPARE	MXAPIASSETSPARE	Read
MXSTORELOC	MXAPILOCATION	Read
MXAPIVENDOR	MXAPIVENDOR	Read
INNO_CLASSIFICATION	MXAPICLASSTRUCTURE	Read
MXAPINUMERICDOMAIN	MXAPINUMERICDOMAIN	Read
MXAPIJOBPLAN	MXAPIJOBPLAN	Read
MXAPIFAILURECODE	MXAPIFAILURECODE	Read
MXAPIFAILURELIST	MXAPIFAILURELIST	Read
INNO_GETWO	MXAPIWODETAIL	All
INNO_ASSET	MXAPIASSET	Read, Save
INNO_ASSIGNMENT	MXAPIASSIGNMENT	Read,Save,Insert,Delete
INNO_ATTACHMENT	INNO_ATTACHMENT	Read,Save,Insert,Delete
INNO_LABTRANS	INNO_LABTRANS	Read,Save,Insert,Delete
INNO_USER	INNO_USER	Read
INNV_WORKLOG	MXAPIWORKLOG	Read,Save,Insert,Delete
MXAPIINSPRESULT	MXAPIINSPRESULT	Read,Save,Insert,Delete
MXAPIINVENTORY	MXAPIINVENTORY	Read
INNO_RELATEDRECORD	INNO_RELATEDRECORD	Read

4. On the **Users** tab, add the integration user created in [Create the Maximo Integration User \(on page 58\)](#).

Figure 4-41 Integration user on the Users tab of INNO_MAXUSER



4.2.14. Activate Maximo Kafka Cron Tasks

Activate the Maximo cron tasks that forward queued messages to Kafka and process inbound error messages.

These are Maximo cron tasks. They are separate from the iMaintenance application scheduled jobs.

1. Open the **Cron Task Setup** application.

2. Open the **Kafka** cron task and set up the following instances.

Cron Task	Description	Access Level
Kafka	Kafka cron task	FULL

kafkacqout

Cron Task Instance Name	Schedule	Run as User	Active	Keep History
kafkacqout	10s,*****	MAXADMIN	Y	Y

Parameter	Value	Description
MESSAGEHUB	KAFKA	Kafka message hub name
MESSAGEPROCESSOR	psdi.iface.jms.QueueToDestManagerProcessor	Kafka message processor class
PARTITION	0	Kafka queue partition
TOPIC	CQOUT	Kafka queue topic name

kafkasqout

Cron Task Instance Name	Schedule	Run as User	Active	Keep History
kafkasqout	30s,*****	MAXADMIN	Y	Y

Parameter	Value	Description
MESSAGEHUB	KAFKA	Kafka message hub name
MESSAGEPROCESSOR	psdi.iface.jms.QueueToDestManagerProcessor	Kafka message processor class
PARTITION	0	Kafka queue partition
TOPIC	SQOUT	Kafka queue topic name

Figure 4-42 Kafka cron task (schedules are masked in the source; use the values in the table)

Cron Task

Cron Task

Kafka

Kafka cron task

Class

psdi.iface.jms.MIFKafkaCronTask

Access Level

FULL

Cron Task Instances

(1 - 2 of 2)

Cron Task Instance Name

Schedule

Run as User

Active?

Keep History?

Max Number of History Records

kafkacqout

30s,*****

MAXADMIN

20

kafkasqout

30s,*****

MAXADMIN

20

Parameters

History

Cron Task Parameters

(1 - 4 of 4)

Parameter

Value

Description

MESSAGEHUB

KAFKA

Kafka message hub name

MESSAGEPROCESSOR

psdi.iface.jms.QueueToDestManagerProcessc

Kafka message processor class

Innovapptive Documentation

151

3. Open the **KAFKAERROR** cron task and set up the following instance.

Cron Task	Description	Access Level
KAFKAERROR	Kafka error message cron task	FULL

kafkaerrorcqin

Cron Task Instance Name	Schedule	Run as User	Active	Keep History
kafkaerrorcqin	30s,*****	MAXADMIN	Y	Y

Parameter	Value	Description
MESSAGEPROCESSOR	psdi.iface.jms.QueueToMaximo-Processor	Kafka message processor class
TOPIC	CQINERR	Kafka queue topic name

Figure 4-43 KAFKAERROR cron task

Cron Task

Cron Task: KAFKAERROR (Kafka error message cron task)

Class: psdi.iface.jms.MIFKafkaErrorProcessingCronTask

Access Level: FULL

Cron Task Instances (1 - 1 of 1)

Cron Task Instance Name	Schedule	Run as User	Active?	Keep History?	Max Number of History Records
kafkaerrorcqin1	30s,*****	MAXADMIN	Active	Keep History	1,000

Parameters (1 - 2 of 2)

Parameter	Value	Description
MESSAGEPROCESSOR	psdi.iface.jms.QueueToMaximoProcessor	Kafka message processor class
TOPIC	CQINERR	Kafka queue topic name

4. Activate the cron task instances and reload the cron tasks.

4.2.15. Configure the iMaintenance External System

Bind the iMaintenance publish channels to their Kafka endpoints in Maximo.

1. In the **External Systems** application, create the following external system.

System	Description	End Point	Enabled
iMaintenance	Custom External system to sync the MAS data with i-Maintenance		Y

2. Set the queues.

Field	Value
Outbound sequential queue	SQOUT
Outbound continuous queue	CQOUT
Inbound continuous queue	CQIN

Figure 4-44 Queues of the iMaintenance external system

System | Publish Channels | Enterprise Services

System: iMaintenance | Description: Custom external system to sync the MAS data with iMaintenance | Enabled? ☒

End Point: | Bidi Configuration: | Text Message Encoding:

Outbound Sequential Queue: | Outbound Continuous Queue: | Inbound Sequential Queue: | Inbound Continuous Queue:

Properties for End Point (0 - 0 of 0)

Property	Value	Encrypted Value
There are no rows to display.		

3. On the **Publish Channels** tab, add the following publish channels.

Publish Channel	Description	Adapter	End Point	User Defined	Use Continuous Queue	Enabled
INNO_ASSET	Custom PS to send the data to iMaintenance	MAXIMO	INNO_-MASTER	Y	Y	Y
INNO_-ASSIGNMENT	Custom Publish Channel to send out WO's Assignment	MAXIMO	INNO_WO	Y	Y	Y
INNO_ATTACHMENT	Innovapptive Custom PS to Publish Channel to push out Attachment	MAXIMO	INNO_WO	Y	Y	Y
INNO_LAB-TRANS	Custom PS to send the Lab-trans data to iMaintenance	MAXIMO	INNO_WOENTITIES	Y	Y	Y
INNO_LOCATION	Custom PS to update	MAXIMO	INNO_-MASTER	Y	Y	Y
INNO_WO	Custom Publish channel to push out the WO change	MAXIMO	INNO_WO	Y	Y	Y
INNO_WOACT	Custom Publish channel to push out the WO change specially Activities	MAXIMO	INNO_WO	Y	Y	Y
INNV_WORK-LOG	Custom PC to send Work logs update to iMaintenance	MAXIMO	INNO_WOENTITIES	Y	Y	Y
INNO_RELATEDRECORD	Custom PC to send Related Record update to iMaintenance - Innovapptive	MAXIMO	INNO_WOENTITIES	Y	Y	Y

Figure 4-45 Publish channels of the iMaintenance external system

4. Save the external system.

Result

Maximo publishes asset and location changes to `MasterDataTopic`, work order, activity, assignment, and attachment changes to `WorkOrderTopic`, and labor, work log, and related-record changes to `WorkOrderEntitiesTopic`.

4.3. Connect iMaintenance to Maximo

Set up the network paths between Maximo, Kafka, and iMaintenance.

Path	From	To	Protocol	Credentials
Maximo events	Maximo (MAS)	<code><bootstrap-serv- er>:<port></code>	Kafka over TLS (<code>SASL_SSL</code> , <code>SCRAM- SHA-512</code>)	Kafka user and cer- tificate from the In- novapptive infra- structure team
Maximo API calls	iMaintenance	<code><maximo-base-url></code>	HTTPS	Maximo API key of the integration user

Allow the Kafka connection from Maximo to the bootstrap server and port in the customer network. This is a TCP connection; the AWS WAF IP allowlisting in [Control Network Access to iMaintenance \(on page 9\)](#) applies to HTTPS traffic to iMaintenance.

4.4. Expose and Secure Maximo APIs

Secure the Maximo API access that iMaintenance uses with the integration user API key and object structure security.

- iMaintenance authenticates to Maximo with the API key of the integration user. Create the key in [Create the Maximo Integration User \(on page 58\)](#).
- The object structures that iMaintenance reads are secured by authorization name. The `INNO_MAXUSER` group grants the access. See [Configure Maximo Roles and Authorizations \(on page 146\)](#).
- Store the API key as a secret. Do not share it in documents or email.

A Maximo API key is issued by Maximo for calling Maximo. It is not interchangeable with an Innovapptive API key.

4.5. Verify the Maximo Connection

Check the Maximo integration configuration and complete the connection tests supplied for the deployment.

Configuration readiness checklist

- The custom attributes, relationships, and condition exist, and the database configuration is applied.
- The integration user exists, has an API key and default site, and is in `INNO_MAXUSER`.
- The twelve object structures and their queries exist.
- The integration controls and the nine iMaintenance publish channels exist, with their processing rules enabled.
- The `INNO_MASTER`, `INNO_WO`, and `INNO_WOENTITIES` endpoints point to the supplied Kafka bootstrap server and topics.
- The Kafka certificate is imported, the `KAFKA` message provider is active, and the four queues exist.
- The automation scripts and launch points are active.
- The `kafkacqout`, `kafkasqout`, and `kafkaerrorcqin` cron task instances are active.
- The `iMaintenance` external system is enabled with the nine publish channels.

A saved configuration does not prove end-to-end delivery. Run the connection tests for the deployment.

5. Initialize iMaintenance

Initialize iMaintenance data, run backend data ingestion and field mapping, and prepare application services and scheduled jobs.



Note:

Complete network access and the SAP or Maximo connection first: data ingestion needs the backend APIs to be active and reachable. Then run these steps in the order listed. Skipping or misordering tasks can result in broken integrations, failed API calls, or missing configurations.

Related tasks

- [Verify Prerequisites \(on page 157\)](#): confirm tools, credentials, and seed data sources
- [Clear Existing MongoDB Collections \(Optional\) \(on page 158\)](#): optional; only when reinitializing an existing environment or switching SAP systems, before you seed data again
- [Seed MongoDB with Core Data \(on page 158\)](#): load JSON configurations, run the seed script, and validate collections
- [Insert SAP Data into MongoDB \(on page 160\)](#): run inbound services and confirm integration statuses
- [Map Dependent Fields \(on page 161\)](#): align Plant and Asset references in MongoDB
- [iMaintenance Application Services \(on page 162\)](#): identify the services that power iMaintenance
- [iMaintenance Scheduled Jobs \(on page 163\)](#): enable background sync and asset refresh tasks
- [Set Up Platform Alerts and IMES for Shift Handover \(on page 164\)](#): if your deployment uses Shift Handover Reports with platform Alerts or IMES

Insert SAP Data into MongoDB and Map Dependent Fields apply to SAP systems.

5.1. Verify Prerequisites

Check backend connectivity, database access, seed files, repositories, and tools needed to initialize iMaintenance.

- **Backend Connectivity** → Verify that the backend APIs are active and reachable from your network. If iMaintenance connects to SAP, these are the required SAP APIs. If it connects to Maximo, complete [Connect to Maximo \(on page 53\)](#) first.
- **Seed Data Access** → Confirm access to the latest JSON seed files from the *devdb* or provided Innovapptive repositories.
- **System Tools Installed** → Ensure the following tools are available on the host where setup will run:

- Node.js and NPM
 - Access to MongoDB Atlas
 - Access to AWS S3 console
 - Access to project repositories (RACE backend, seed data folders)
- At the end of this step, you can confidently source seed data and connect to your backend system and MongoDB without dependency issues.

5.2. Clear Existing MongoDB Collections (Optional)

Clear existing MongoDB collections when reinitializing the environment or switching SAP systems.

A clean start prevents conflicts with outdated data when you reseed.

This procedure uses the SAP inbound utilities and applies to SAP deployments.

1. Before you run the script, make sure the `nosql` object in the script contains valid database credentials and points to the database you intend to clear.
2. Navigate to the reset script: `RACE/backend/SapInbound/src/utilities/deleteCollections.js`
3. Run the script: `node deleteCollection.js`

5.3. Seed MongoDB with Core Data

Load the core configuration, role, permission, form, and reporting data into MongoDB.

The seed data gives iMaintenance its baseline configuration: roles, forms, dashboards, preferences, and integration templates. Without it, the application launches with missing or incomplete data.

To seed MongoDB with core data:

1. Source Latest JSON Files

- Location: **RACE/backend/Seed Data/JSON**
- Retrieve the latest configuration, permission, integration, and form-related JSONs from *devdb*.
- If configurations are not coming from SAP, include the default configuration for **attachments** and ensure **Plant** is added manually in the DB.

2. Verify Required JSON Files

- Integrations
- SAP Inbound (SAP deployments only)
- AlertString
- Configurations
- Dashboard
- Language (Usecase – Language)
- MasterBo (Reports)
- MasterForm, MasterFormPage, MasterFormPageSection, MasterFormPageSectionField, MasterFormPageSectionFieldLogic
- MobileAppPreference
- Permission
- Report
- RiskMatrix
- Role
- User
- Widget, WidgetConfiguration

3. Confirm Storage Locations

- **AWS S3 Bucket:** **race2.0resources**
- **Drive Link:** **RACE2.0 SEED DATA** (Google Drive folder)

4. Update MongoDB Connection Configuration

- File Path: `RACE/backend/Seed Data/connection/mongo.js`
- Modify the `nosql` object with the appropriate values:
 - Database
 - Host (only if switching MongoDB clusters)
 - Username
 - Password

5. Run Seed Script and wait for the process to complete.

```
cd "RACE/backend/Seed Data"
node index.js
```

Result

Wait for the process to complete. Seed data is now inserted into MongoDB.

5.4. Insert Seed Data

Seed data is inserted by the seed script in Seed MongoDB with Core Data.

Run the seed script once, as described in [Seed MongoDB with Core Data \(on page 158\)](#).

That task includes the command, the wait for completion, and the result.

5.5. Insert SAP Data into MongoDB

Load SAP master and transactional data into MongoDB through the SAP inbound service.

This lets iMaintenance work with real SAP objects instead of placeholders. When the task is complete, the inbound services are active and the collections show the correct status.

To insert SAP Data into MongoDB

1. Update SAP MongoDB Connection

- File Path: `RACE/backend/SapInbound/src/direct_adaptor/lib/mongo.js`
- Modify the `nosql` object with correct values (Database, Host, Username, Password).

2. Start SAP Inbound Microservice

```
cd RACE/backend  
npm run inbound
```

Monitor terminal logs to confirm data ingestion is running.

3. Monitor MongoDB Atlas

- Navigate to the **Integrations** collection.
- Check the `status` field:
 - `0` → Not Active
 - `1` → Active
 - `3` → Running
 - `4` → Failed
- Wait until logs confirm **completion**.

5.6. Map Dependent Fields

Map dependent SAP fields in MongoDB and complete the integration updates needed when switching SAP environments.

Some fields, such as **Plant**, must be mapped in MongoDB so that SAP data and iMaintenance line up. The seed mapping script creates these mappings.

To map dependent fields:

1. Run Seed Mapping Script

- File Path: `RACE/backend/SapInbound/src/integration-v2/startSeed.js`
- `node startSeed.js`

This maps dependent fields (e.g., Plant) correctly in MongoDB.

2. Finalize Mapping

- Stop the SAP Inbound microservice after the script completes.
- If switching SAP environments, update the **Integrations** collection in MongoDB Atlas to reflect the new environment.

5.7. iMaintenance Application Services

Identify the application services used for authentication, SAP integration, maintenance records, forms, reporting, and AI.

The table lists the iMaintenance services and their purpose. Which services a deployment runs depends on its backend system and the features in use.

Key Micro-services and Their Purpose

Micro-service	Purpose
ai-chatbot-service	Digital Work Instruction
ai-localisation-service	Language Translation
asset360-service	Asset 360's Dashboard & Data Prep
alert-api-service	Alerts & Push Notifications
auth-api-service	Authentication
authored-form-api-service	Field Configuration Forms
erp-master-data-api-service	General Dropdowns
erp-outbound-consumer-service	Post Transaction Data to SAP
erp-outbound-producer-service	Listen for Transaction Events
integration-manager-api-service	(Details not specified)
master-configuration-api-service	Plant/Asset/Functional Location Master Data

Micro-service	Purpose
mobile-app-preference-api-service	Global/Transactional Preferences
my-forms-api-service	Form Templates
operator-rounds-api-service	Observations, Issues, Actions
report-api-service	Dashboard Reports
sap-inbound-api-service	Initial SAP Data Ingestion
settings-api-service	General Settings
storage-api-service	Attachment Handling
user-service-api-service	Users, Roles, Permissions, Groups
workorder-api-service	MCC (Work Order Management)
platform-dev → auth-service	oAuth (Google/Microsoft) Authentication
AI Issue Detection Service	Autonomous Issue Detection
AI WorkOrder	Autonomous Work Order Creation

Deployment Notes

- Each microservice is typically packaged as a **Node.js service** (often containerized).
- They should be started and monitored individually or through your orchestration platform.
- Validate that each service your deployment uses is running before proceeding to cron jobs and higher-level configurations.

5.8. iMaintenance Scheduled Jobs

Identify the jobs that synchronize SAP updates and refresh Asset360 data, and the messaging and cache services they depend on.

These are iMaintenance application jobs. For Maximo, the Maximo Kafka cron tasks are set up separately in [Activate Maximo Kafka Cron Tasks \(on page 149\)](#).

Scheduled jobs keep iMaintenance current without manual work, for example by syncing incremental updates and refreshing asset dashboards.

Cron Jobs

- **delta-cronjob** → Synchronizes incremental updates from SAP (SAP deployments only).
- **asset360-cronjob** → Refreshes Asset360 dashboard data.

Supporting Components

- **Messaging Queue** → `rabbitmq-service` handles asynchronous communication.
- **Distributed Caching** → `redis-cluster` maintains high-speed cache for frequently accessed data.
- **CWP Portal** → `race-2-client` connects the cron jobs and microservices with the portal.

Deployment Notes

- Use standard Linux `crontab` to schedule these jobs at required intervals.
- Validate execution logs after the first run to confirm jobs are firing correctly.
- Ensure RabbitMQ and Redis are up and running before enabling cron jobs.

5.9. Set Up Platform Alerts and IMES for Shift Handover

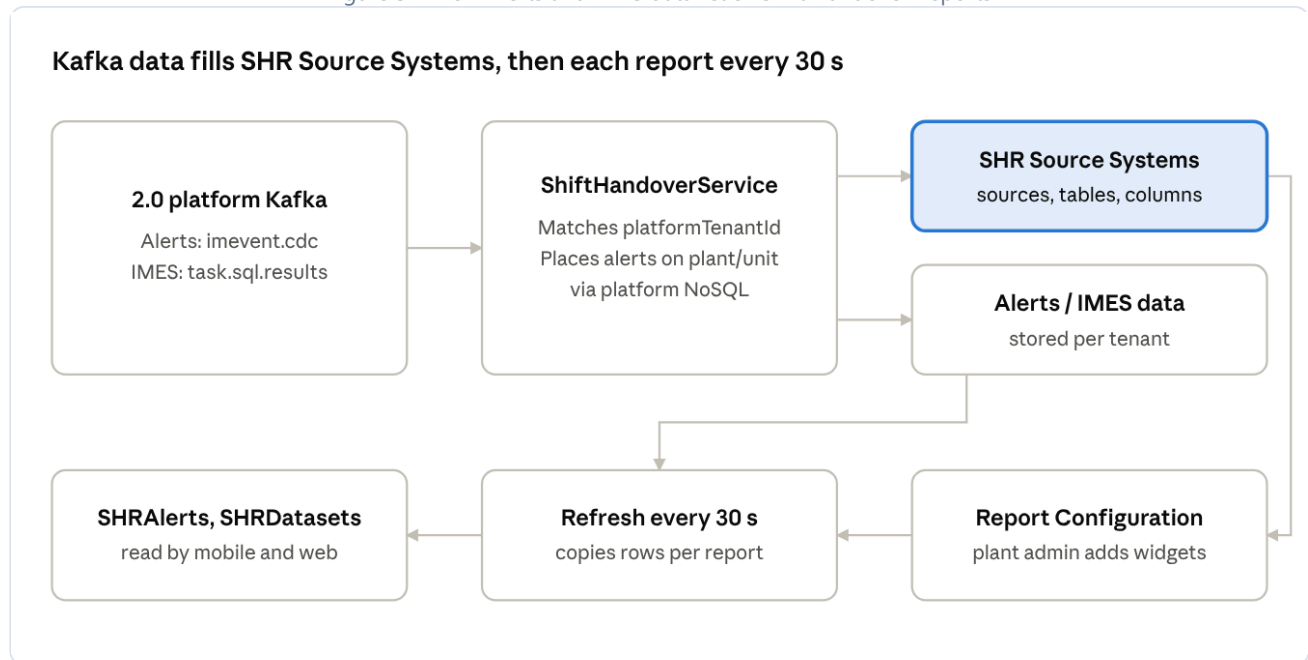
Prepare the tenant catalog and service settings that bring platform Alerts and IMES results into Shift Handover Reports.

Before you begin

- Redis, MySQL, and `ENCRYPTION_KEY` are present for ShiftHandoverService.
- The 2.0 platform team has supplied the platform tenant ID, the platform MongoDB connection details, and the Kafka broker list.
- For IMES: the iMES platform services (task-orchestration-api and temporal-worker) are running and publish IMES results to the Kafka topic `task.sql.results`. Platform administrators set them up with the separate iMES Platform Deployment and Administration runbook.

If your deployment uses Shift Handover Reports with platform Alerts or IMES, set up the tenant so that platform Alerts and IMES results reach Shift Handover Reports. The DevOps team does steps 1 to 5. A plant admin does step 6.

Figure 5-1 How Alerts and IMES data reach Shift Handover Reports



1. Run the catalog migration. It adds the new catalog columns, the feature flags, and the tenant configuration defaults, and keeps values already set.
2. Turn on the feature flags under `configurations.featureFlags` in the tenant catalog.

```
featureFlags": {  
  "enableSHRAAlerts": true,  
  "enableSHRIMES": true,  
  "enableSHRCustomSections": true,  
  "enableSHRAAlertsUnified": false  
}
```

For what each flag does, see *Configure Shift Handover Alerts and IMES Settings in the iMaintenance Configuration Guide* or *Configure Shift Handover Alerts and IMES Settings in the mRounds Integration and Configuration Guide*. Keep `enableSHRAAlertsUnified` false until the unified Alerts view is available for the deployment. The flag works only on the unified platform, not in standalone CWP.

3. Fill the tenant configurations next to `featureFlags` : `shrAlertConfig` , `shrBuilderLimits` , and `imesConfig` .

```
"shrAlertConfig": {  
  "severityColors": { "critical": "#D92D2D", "major": "#FFAB40", "medium": "#4DC7E7", "minor": "#8F83A3" },  
  "statusColors": { "open": "#D92D2D", "in-progress": "#4F73FE", "closed": "#12B76A" },  
  "topN": 100  
},  
"shrBuilderLimits": {  
  "maxSections": 5,  
  "maxWidgetsPerSection": 5  
},  
"imesConfig": {  
  "baseUrl": "https://<platform-host>/task-orchestration"  
}
```

Keep `topN` at 100 unless a requirement calls for more; a larger list can exceed the 240 KB mobile sync limit. If `imesConfig.baseUrl` is empty, IMES refresh at report freeze is skipped; streamed IMES data is still shown.

4. Fill `platformTenantId` and `nosqlPlatform` on the tenant row of the `catalogs` table. Get both values from the 2.0 platform team.

Column	Type	Example	Used for
<code>platformTenantId</code>	Text, up to 100 characters	qa-cwp-hana	Matches each Kafka message to this tenant. Alerts carry it in tenant; IMES results carry it in tenantId. Also sent to TaskOrchestration on IMES refresh.
<code>nosqlPlatform</code>	JSON	See the sample below	Connection to the shared platform MongoDB, where alerts are mapped to a CWP plant, unit, asset, and position through its TaxonomyNode collection.

```
nosqlPlatform: {  
  "prefix": "mongodb+srv",  
  "host": "<platform-cluster-host>",  
  "port": 27017,  
  "database": "<platform-db-name>",  
  "user": "<platform-db-user>",  
  "password": "<encrypted, same scheme as the nosql column>"  
}
```

- The fields are the same as the tenant's existing `nosql` column. Store the password encrypted, never in plain text.
- `prefix` is `mongodb` (uses host and port) or `mongodb+srv` (uses host only).
- If `platformTenantId` is empty, messages are matched by tenant domain instead (for example, qa-cwp-maximo). Set it anyway; domain matching is a fallback.
- If `nosqlPlatform` is empty, alerts are stored but have no plant, so they appear on no report.
- An alert maps to a plant only when its platform taxonomy node carries a DYNAMO external ID that matches a CWP asset, location, or plant.

5. Set the Kafka environment variables on ShiftHandoverService. See [Configure Shift Handover Kafka Consumers \(on page 168\)](#).

6. Hand over to the plant admin, who adds the Alerts and IMES sections and widgets to each plant's report configuration in the CWP web app.

See Add Alerts and IMES Widgets to Shift Handover Reports in the *iMaintenance Configuration Guide* or Add Alerts and IMES Widgets to Shift Handover Reports in the *mRounds Integration and Configuration Guide*.

5.9.1. Configure Shift Handover Kafka Consumers

Enable the Alerts and IMES consumers and keep shared-cluster environments in separate consumer groups.

ShiftHandoverService reads from the Kafka cluster and topics used by the 2.0 platform; nothing new is produced on the CWP side. Both consumers are off until enabled. These settings are separate from the Maximo Kafka endpoints in [Connect to Maximo \(on page 53\)](#).

1. Set the following environment variables on ShiftHandoverService.

Variable	Default	Set to
KAFKA_BROKERS	none	The 2.0 platform team's broker list, comma-separated. Required.
IMEVENT_KAFKA_ENABLED	false	true to receive Alerts
IMEVENT_KAFKA_TOPIC	imevent.cdc	Keep unless the 2.0 platform team uses another name
IMES_KAFKA_ENABLED	false	true to receive IMES results
IMES_KAFKA_TOPIC	task.sql.results	Keep unless the 2.0 platform team uses another name
KAFKA_GROUP_ENV	unset	A name unique to each environment and identical on all its pods (for example, the Kubernetes namespace). Required when several environments share one Kafka cluster.
KAFKA_TOPIC_GROUPS	derived	Leave unset. Override format only: topic:group,topic:group

2. If several environments share the Kafka cluster, set `KAFKA_GROUP_ENV` to a value unique to the environment, for example from the Kubernetes namespace.

```
env:
- name: KAFKA_BROKERS
  value: "b-1.platform-kafka:9092,b-2.platform-kafka:9092"
- name: IMEVENT_KAFKA_ENABLED
  value: "true"
- name: IMES_KAFKA_ENABLED
  value: "true"
- name: KAFKA_GROUP_ENV
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
```

Example broker addresses are illustrative. Use the broker list supplied by the 2.0 platform team.

3. Leave the other Shift Handover variables unset. The refresh interval (30 seconds), `SHR_ALERT_WINDOW_HOURS` (48), batch sizes, and cron schedules have working defaults.

Result

With `KAFKA_GROUP_ENV=qa-cwp-maximo`, the consumer groups are `CBO_SHIFT_HANDOVER-qa-cwp-maximo-imevent.cdc` and `CBO_SHIFT_HANDOVER-qa-cwp-maximo-task.sql.results`. Without it, environments on the same cluster split the messages, and each sees only part of the data.

6. Troubleshoot Deployment Issues

Resolve the documented SAP OData metadata issue and trace missing Shift Handover Alerts and IMES data.

The documented issue applies to SAP deployments. One issue is documented: OData metadata that does not populate when the API is called through SAP API Management. See [Resolve Missing OData Metadata in S/4HANA \(on page 170\)](#).

For Shift Handover Alerts and IMES, see [Troubleshoot Shift Handover Alerts and IMES \(on page 171\)](#).

6.1. Resolve Missing OData Metadata in S/4HANA

Apply and test the API proxy header policy used for the documented OData metadata issue.

In SAP HANA-based systems, you may encounter issues where OData metadata does not populate when the API is invoked.

Cause

This error occurs when the backend system compresses metadata responses using the Accept-Encoding header. Some versions of HANA do not handle decompression properly, causing the metadata fetch to fail.

Resolution

Use the **Assign Message** policy in the API Proxy to override the Accept-Encoding header.

1. Select **API proxy** and click **Policies**.
2. Click **Edit**.
3. Select **ProxyEndpoint**.
4. Select **PreFlow**.
5. Under **Mediation Policies**, choose **Assign Message**.
6. Click on the + button to add the policy.
7. Enter the **Policy Name**, set the Stream as Incoming Request and click **Add**.
8. Replace the existing default content in the Body section with the following policy XML, and click **Update**.
9. **Policy Message:**

```
<!-- This policy can be used to create or modify the standard HTTP request and response messages -->
<AssignMessage async="false" continueOnError="false" enabled="true" xmlns="http://www.sap.com/apimgmt">

  <!-- Sets a new value to the existing parameter -->
  <Set>
    <Headers>
      <Header name="Accept-Encoding">gzip,deflate</Header>
    </Headers>
  </Set>
  <IgnoreUnresolvedVariables>false</IgnoreUnresolvedVariables>
  <AssignTo createNew="false" type="request"></AssignTo>
</AssignMessage>
```

10. Click **Save**.

11. Now, the policies have been successfully defined. Proceed to deploy the API proxy to apply the changes.

6.2. Troubleshoot Shift Handover Alerts and IMES

Trace incoming messages through tenant matching, plant mapping, and report refresh to find missing data.

A working setup shows new alerts or IMES rows on the open report within about 30 seconds. If it does not, check the ShiftHandoverService logs in this order.

Check	Log line to look for	If missing or wrong
Message received	[imEvent] <- received ... (alerts) or [imes] <- received ... (IMES)	Consumer not enabled, wrong KAFKA_BROKERS, or another environment in the same consumer group (set KAFKA_GROUP_ENV)
Tenant matched	No "no tenant in the catalog for domain ..." error	platformTenantId empty or different from the tenant in the message
IMES allowed	No "enableSHRIMES is off for ..." message	Turn on enableSHRIMES
Alert placed on a plant	[imEvent] scope for ... plant=<id>	plant=null: nosqlPlatform missing, or the taxonomy node has no DYNAMO ID for a CWP asset or location
Reports marked	[imEvent] -> marked N SHR(s) dirty, with N greater than 0	The alert is outside the shift or more than 48 hours from now, or the plant report level is off
Reports refreshed	[shrAlertsWorker] <tenant>: N dirty -> {"written": ...}	SHR_ALERTS_WORKER_ENABLED is set to false

Common surprises

- **Widget shows nothing on today's report:** the widget was added after the report was generated. Wait for the next shift's report.
- **Source missing in Create Widget:** no message from that source has arrived yet, or its flag (`enableSHRAAlerts` or `enableSHRIMES`) is off.
- **Alerts tab missing:** `enableSHRAAlerts` is off, or the Alerts section is not selected in the plant's configuration.