



# Infor ION Desk User Guide—Cloud Edition

2020-x

### **Important Notices**

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

### **Trademark Acknowledgements**

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

### **Publication Information**

Release: Infor ION 12.0.x

Publication Date: May 7, 2020

Document code: ion\_12.0.x\_iondeskceug\_en-us

# Contents

<b>About this guide.....</b>	<b>16</b>
Contacting Infor.....	16
<b>Chapter 1: Infor ION introduction.....</b>	<b>17</b>
ION Connect.....	18
Event Management.....	18
Alarms.....	19
Workflow.....	19
Activities.....	20
ION homepages widgets.....	20
<b>Chapter 2: ION Desk overview.....</b>	<b>23</b>
Home page.....	26
Model overview.....	26
Tile toolbar.....	27
Single-select mode.....	27
Multi-select mode.....	28
Grids overview.....	29
Details page.....	29
Indicators.....	30
Bookmarking pages.....	31
Starting ION Desk.....	31
Viewing ION Desk version details.....	32
<b>Chapter 3: ION Connect concepts.....</b>	<b>33</b>
Documents.....	33
Connection points.....	35
Data flows.....	35
Document flows.....	36

---

Data Lake flows.....	36
Compare data flows and workflows.....	37
Message Routing.....	39
Examples for using explicit routing on reply messages.....	40
XML document from external source.....	41
<b>Chapter 4: ION Connect Modeling.....</b>	<b>42</b>
Prerequisites.....	42
Using the modeler.....	42
Creating and using data flows.....	43
Modeler toolbox.....	43
Using the notifications pane.....	47
Activating and deactivating data flows.....	47
Creating and using API flows.....	49
Using the API Flow modeler.....	49
API Flow parameters.....	50
Publishing.....	51
API Flow elements.....	51
API Flow Properties.....	52
ION API step.....	54
Parallel API flows.....	56
Decision API flows.....	56
Monitoring.....	58
Error handling.....	59
Creating and using connection points.....	59
Creating connection points from the data flow modeler.....	59
Creating connection points from the connection points list.....	60
Changing a connection point.....	61
Checking usage of connection points.....	61
Deleting a connection point.....	62
File (SFTP from Cloud) connection point.....	62
Enterprise Connector.....	63
Integration with your local applications.....	63
Downloading and installing the Enterprise Connector.....	65
Configuring Enterprise Connector.....	70
Managing the Grid by REST services.....	75

---

Updating the JDK version.....	76
Updating location credentials.....	76
Log archiving.....	77
Configuring the use of a proxy service.....	77
Scale-out the Enterprise Connector to another host.....	78
Uninstalling an Enterprise Connector host.....	79
Enterprise Connector service - processing.....	79
Upgrading Enterprise Connector from an earlier 12 version.....	81
Disaster recovery planning.....	83
Application connection points.....	84
Configuring an IMS integration through ION API.....	84
General application properties.....	86
Network connection point.....	91
Configuring a Network connector.....	91
Tenant – Tenant dependency.....	92
Document types supported by various connectors.....	92
Document mappings.....	92
Splitter.....	93
Using a splitter in a data flow.....	93
Rules for using a splitter in a data flow.....	94
Workflow from document flow.....	97
Modeling a workflow from document flow.....	98
Workflow from document flow process.....	100
Filtering and content-based routing.....	101
Filtering.....	102
Routing.....	103
Selecting attributes.....	103
Defining conditions.....	105
Attribute-value comparison.....	106
Attribute comparison.....	107
Attribute existence.....	108
Combined condition.....	108
Advanced properties of connection points.....	111
Application connection points of type IMS.....	111
Advanced properties for polling.....	111

---

Protected content.....	113
Defining protected content.....	113
Exporting content as protected.....	114
<b>Chapter 5: ION Connect Mapping.....</b>	<b>115</b>
Concepts.....	116
Graphical mapping.....	116
Function.....	116
Smart Matching.....	117
XSLT Generator V1 and V2.....	118
XSLT and advanced mapping.....	119
Sensitive Data Mapping.....	120
Creating, changing, or deleting mappings.....	120
Creating mapping from the data flow modeler.....	120
Creating graphical mapping from the Mappings menu.....	120
Creating a sensitive data mapping from the Mappings menu.....	121
Changing a mapping.....	122
Deleting a mapping.....	123
Using the graphical mapping modeler.....	123
Defining a graphical mapping.....	123
Document tree.....	124
Modifying the working area of the Mapper window.....	124
Tree navigation.....	124
Searching and filtering.....	124
Focus view.....	126
Graphical mapping connections.....	127
Set default value.....	129
Grouping.....	130
Error and warning notifications.....	133
Graphical mapping data type compatibility.....	135
Element occurrence compatibility.....	136
Graphical mapping using functions.....	139
Starting points of functions.....	140
Using function scopes.....	142
Adding constants or user-defined functions.....	143
Connecting functions.....	149

---

Single and List functions.....	151
Loop constructs (xsl:for-each).....	153
Removing a function.....	153
Advanced mapping.....	153
Using advanced mapping.....	153
Importing and exporting XSLT.....	154
Editing XSLT.....	155
XSLT extensions.....	156
Switching back to graphical mapping.....	162
Sensitive data mapping security policies.....	162
Creating a security policy.....	163
Changing applied security policy.....	163
Creating and updating security policy on multiple nodes.....	163
Applying the Data Masking security policy.....	164
Applying the Data Removal security policy.....	164
Applying the Node Removal security policy.....	164
Security policy status and feedback.....	165
Removing a security policy for a node.....	165
Testing a mapping.....	166
Comparison tool.....	166
Icons and visual feedback.....	166
Versioning.....	170
Version administration.....	170
Approving draft versions.....	171
Version and approval status in the Mappings overview page.....	171
<b>Chapter 6: ION Scripting.....</b>	<b>173</b>
ION Scripting concepts.....	173
Supported languages.....	174
Supported libraries.....	174
Scripts overview page.....	175
Creating scripts.....	175
Deleting scripts.....	175
Duplicating scripts.....	176
Exporting scripts.....	176
Importing scripts.....	176

---

Script details.....	177
Approving scripts.....	177
Script usage.....	177
Searching and filtering scripts.....	177
Libraries overview page.....	178
Importing libraries.....	178
Deleting libraries.....	179
Exporting libraries.....	179
Library details.....	179
Library usage.....	179
Searching libraries.....	180
Libraries.....	180
Information tab.....	181
Browsing the library content.....	181
Library versioning.....	182
Finding and creating libraries.....	182
Script modeling.....	182
Input and output variables names.....	183
Writing the code.....	184
Library dependencies.....	185
Testing a script.....	186
Test data syntax-highlight and pretty-print.....	187
Script documentation.....	187
Script versioning.....	188
Using scripting.....	189
Scripts in data flows.....	189
Limitations.....	189
Runtime errors and exceptions.....	190
<b>Chapter 7: ION Connect import and export.....</b>	<b>191</b>
Import procedure.....	191
Generic import procedure.....	192
Mapping specific behavior.....	192
Data flow and connection point specific behavior.....	193
Export procedure.....	193



---

<b>Chapter 8: Event Management.....</b>	<b>195</b>
Architecture.....	195
Examples.....	196
Concepts.....	197
Monitors.....	197
Application documents.....	198
References.....	199
Attributes.....	200
Monitor rules.....	202
Procedures.....	224
Task overview.....	224
Modeling tasks.....	225
<b>Chapter 9: Alarms.....</b>	<b>237</b>
Architecture.....	237
Examples.....	238
Concepts.....	239
Alarm templates.....	239
Alarms.....	240
Procedures.....	240
Task overview.....	240
Modeling tasks.....	241
<b>Chapter 10: Workflow.....</b>	<b>246</b>
Working with a set of workflow definitions.....	246
Architecture.....	247
Starting a workflow.....	247
Workflow definition.....	249
Sample workflow.....	250
Concepts.....	251
Workflow patterns.....	251
Workflow elements.....	252
Workflow parameters.....	255
Workflow conditions.....	257
Workflow drillbacks.....	261
Workflow tasks.....	261

---

Workflow structures.....	264
Workflow ION API call.....	265
Procedures.....	272
Task overview.....	272
Creating a workflow definition.....	273
Activating workflow definitions.....	310
Changing active workflow definitions.....	310
Deactivating workflow definitions.....	312
Checking usage of workflow definitions.....	312
Exporting workflow definitions.....	313
Importing workflow definitions.....	313
Workflow authorizations.....	313
Starting workflow instances.....	314
Canceling workflow instances.....	317
Activation policies.....	318
Workflow schedules.....	323
<b>Chapter 11: Monitors and Workflows advanced configuration.....</b>	<b>328</b>
Adding translations.....	328
Configuring escalation and reminders.....	328
Adding distribution elements.....	331
Adding distribution elements in simple configuration mode.....	332
Adding distribution elements in the advanced configuration mode.....	334
<b>Chapter 12: Business Rules.....</b>	<b>337</b>
Basic navigation.....	337
Permissions.....	338
Permissions per role.....	338
Business Rules matrices.....	339
Lifecycle of a matrix.....	339
Procedures.....	340
Approval matrix definition.....	340
Decision matrix definition.....	342
Defining matrix parameters.....	344
Defining the matrix rules.....	345
Overview of the condition types and their syntax.....	346

---

Editing the actions for an approval matrix.....	348
Editing the actions for a decision matrix.....	351
Adding notes to matrix rows.....	352
Submitting a matrix for approval.....	352
Duplicating a matrix definition.....	352
Reverting to the last active version.....	352
Importing and exporting matrix definitions.....	353
Importing and exporting matrix definitions from and to CSV.....	353
Viewing matrices.....	355
Viewing previous matrix versions.....	356
Approving and activating a matrix.....	357
Simulating the matrix execution.....	357
Using an approval matrix from Workflow.....	357
Using a decision matrix from Workflow.....	358
Workflow and matrix versions.....	359
Page overviews.....	359
Matrices.....	359
Approval Matrix.....	360
Decision Matrix.....	360
<b>Chapter 13: ION OneView, management and troubleshooting.....</b>	<b>361</b>
ION OneView.....	361
Search documents in ION OneView.....	362
Timeline view search results for documents.....	371
Advanced view search results.....	385
Manage ION Connect.....	386
Active connection points.....	386
Pausing message processing of connection points.....	388
Viewing or deleting pending messages.....	391
Resending messages.....	391
Active data flows.....	392
Error Messages.....	393
Active Routings.....	396
Manage ION Process.....	396
Navigation.....	396
Manage Event Management.....	401

---

Manage Alarm Templates.....	404
Manage ION Workflow.....	407
Manage workflow activation policies.....	411
Manage workflow schedules.....	412
Manage alarms.....	413
Viewing search results.....	414
Viewing alarm details.....	415
Stopping an alarm.....	415
Pulse activities.....	416
Concepts.....	416
Management.....	417
Configurations for ION Pulse.....	427
Troubleshooting and logging.....	435
ION status overview.....	435
ION OneView.....	436
Missing data.....	436
Incorrect data.....	437
Exception handling.....	437
Receiving e-mails for exceptions.....	438
Receiving alerts for business process errors.....	438
Receiving alerts for exceptions.....	439
<b>Chapter 14: Configuration.....</b>	<b>440</b>
Codes.....	440
Creating codes in ION.....	440
Importing codes.....	441
Exporting codes to XML.....	442
Configuring the ION Service.....	442
Properties tab.....	443
Configuration Files tab.....	443
Key Management tab.....	444
Downloads.....	445
Downloading a feature asset.....	445
Exporting the complete history log.....	446
<b>Chapter 15: Data catalog.....</b>	<b>447</b>

---

Object schemas.....	447
Tiles overview.....	447
Grid overview.....	449
Filter overview.....	450
Schema generation wizard.....	451
Adding objects of type ANY.....	465
Importing objects.....	465
Exporting objects.....	467
Deleting objects.....	468
Viewing Object Metadata.....	468
User area extensions.....	475
Importing user area schema files.....	476
Exporting user area schema files.....	477
Deleting user area schema files.....	477
Mapping user area extensions.....	477
Removing user area extensions mappings.....	478
Editing user area extensions mappings.....	478
Locale selections.....	478
Adding locale selections.....	479
Editing locale selections.....	479
Deleting locale selections.....	480
Repositioning locale selections.....	480
Importing locale selections.....	480
Exporting local selections.....	481
<b>Chapter 16: Data Lake.....</b>	<b>482</b>
Data Lake overview.....	483
Storage policies.....	483
Archive criteria.....	484
Setting up storage policies.....	485
Updating a storage policy.....	486
Deleting a storage policy.....	486
Restoring data objects.....	486
Purge data.....	487
Purge by ID.....	488
Purge by filter.....	488

---

Purge logs.....	489
Purge Status Indicators.....	490
Stopping a Purge event.....	490
Compass UI.....	491
Overview.....	491
Open Compass.....	491
Data objects panel.....	492
Data object tree.....	492
Refreshing the list of Data Objects.....	492
Filtering the tree.....	493
Data object menu.....	493
Generating SELECT.....	493
SELECT options.....	494
Generating administrative procedures.....	495
Query editor.....	495
Data Lake widgets.....	499
<b>Chapter 17: ION Desk authorizations.....</b>	<b>500</b>
ION Desk users.....	501
Authorization framework.....	501
Access level.....	501
Standard IFS security roles and desk profiles.....	502
Desk profile administration.....	502
Adding a new IFS security role with a standard profile.....	503
Creating a desk profile.....	504
Duplicating a desk profile.....	504
Modeling role-based access.....	504
Examples.....	505
Importing desk profiles.....	505
Exporting desk profiles.....	506
Specific access behavior.....	506
Document authorization.....	506
OneView documents.....	506
<b>Appendix A: Error Message fields.....</b>	<b>508</b>
<b>Appendix B: Standard functions for ION mappings.....</b>	<b>510</b>

**Appendix C: Tooltips for ION Mapper.....520**

## About this guide

This guide describes how to configure and manage ION Connect, Event Management, Workflow, Data Catalog and Data Lake using ION Desk.

### Intended Audience

The document is intended for this audience:

- System Administrators
- Business Process Administrators
- Database Administrators
- Application Administrators

### Related documents:

You can find the documents in the product documentation section of the Infor Support portal, as described in "Contacting Infor".

- *Infor ION Development Guide*
- The "*Infor Ming.le administration*" section in the *Infor Ming.le Cloud Edition Online Help*
- *Infor ION Technology Connectors Administration Guide*
- *Infor Inbox User Guide*

## Contacting Infor

If you have questions about Infor products, go to Infor Concierge at <https://conciierge.infor.com/> and create a support incident.

The latest documentation is available from [docs.infor.com](https://docs.infor.com) or from the Infor Support Portal. To access documentation on the Infor Support Portal, select **Search > Browse Documentation**. We recommend that you check this portal periodically for updated documentation.

If you have comments about Infor documentation, contact [documentation@infor.com](mailto:documentation@infor.com).

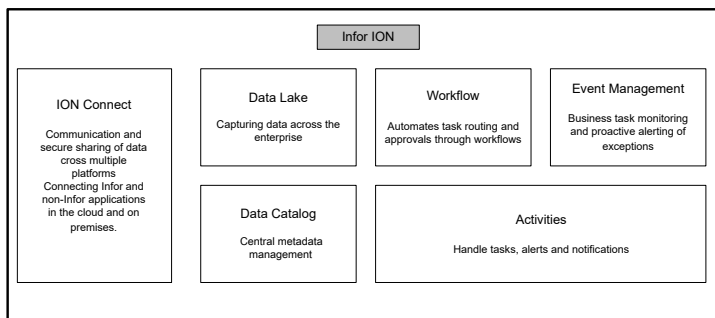


## Chapter 1: Infor ION introduction

Infor ION is an interoperability and business process management platform designed to integrate applications, people, processes and data to run your business.

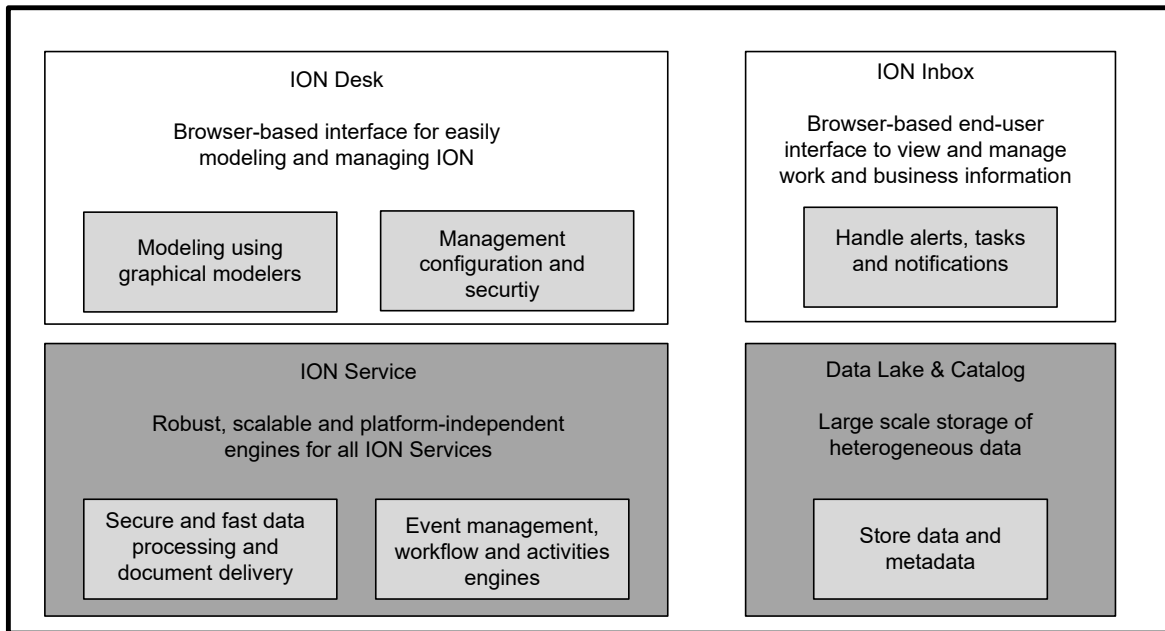
Infor ION enables Infor and customer activities to configure a routed infrastructure, setup workflows, design and activate business event monitors and manage the tasks and alerts through Infor Ming.le™.

This diagram shows the Infor ION components:



The Infor ION suite is a set of services built by Infor that simplifies connectivity and data sharing across all applications.

This diagram shows the Infor ION suite:



## ION Connect

With ION Connect you can establish connections either between applications, which can be Infor applications or third party applications, or between two tenants.

You can configure ION Connect through the ION Desk. The ION Desk is an intuitive, browser-based interface that is used to configure and monitor the ION Service.

In ION Desk, you can model document flows between applications. Such flows can represent a business process. More technical flows can also be defined. For example, mapping data from a third party application to a standard business object document as used by an Infor application. You can also use filtering or content-based routing.

From ION Desk you can deploy your models to ION Service by activating them. ION Service then processes your documents according to your activated models. ION Desk provides management screens to monitor the behavior of ION Service and helps in troubleshooting when required.

## Event Management

In Event Management you can monitor business events and create alerts when exceptions occur.

The start point of monitoring are the messages that are published by applications when business events occur. Monitors are processes that run in the Event Management engine and evaluate these documents by applying pre-defined business conditions.

There is a predefined set of business conditions that the monitoring engine can apply: comparison conditions, value change conditions, and document overdue timer based rules. When exceptions are detected, these are reported as alerts to the business users. The distribution list of alerts is part of the monitor definition.

## Alarms

From ION version 12.0.6 you can model monitoring templates, called “alarm templates”, on the server side. You can use these templates in the Alarms mobile application or Alarms homepage widget to define user-defined monitors, called “alarms”.

The templates contain this configuration information:

- The documents to be monitored
- The attributes from these documents that should be evaluated

The definition of the business conditions is delegated to the end user. The result of alarms which evaluate documents are also alerts, similar to the alerts generated by the monitors from the Event Management module.

Alarms are configured by business users who must monitor a specific event that must occur in a short period of time. To monitor a recurring or generic event, you must create a standard Event Management monitor, that remains active for an indefinite period of time.

## Workflow

Workflow in ION is used to model and run business processes such as authorizations, work distribution, or task-driven workflows across several components.

For authorizations, you can use an approval workflow to approve a purchase order or to sign off a new item. Such workflows are triggered by the creation of a new item or the submission of a new order.

In the area of work distribution, the workflow ensures that all tasks for a certain event are being distributed and executed by different users. The advantage of using the Workflow is that you can model how these tasks are performed, for example sequential or in parallel.

## Activities

Activities is a generic name for Alerts, Tasks and Notifications.

The Pulse engine is the ION component that ensures the communication to the end user. Pulse manages the Alerts, Tasks and Notifications and distributes them to the users. When users take actions on Infor Ming.le activities, Pulse records these actions and returns the reply to the engine that generated the user activity. Pulse is also the component that sends email notifications for Alerts, Tasks and Notifications.

## ION homepages widgets

Using the Homepages application that runs in Infor Ming.le, you can create and configure ION statistics widgets to monitor ION and its services such as Connect or Pulse.

You can use the business process widgets to manage alerts, tasks and notifications and to view the workflow status diagram. You can use the Alarms widget to create user-defined monitors that generate an alert when the business condition is met.

This table shows the ION homepages widgets:

Name	Description
Messages and Events	Real-time counters of all inbound and outbound messages sent to and from ION, and the number of open workflows, tasks, and alerts. This widget replaces the ION Insight widget.
Messages Insight Hybrid	Real-time counters of all inbound and outbound messages sent to and from ION through the Enterprise Connector. This widget replaces the ION Insight Hybrid widget.
Messages by document	Overview of all inbound and outbound messages for a specific document type. This widget replaces the ION by document widget.
Messages by connection	Overview of all inbound and outbound messages for a specific connection point, including statistics on errors and queue size. This widget replaces the ION by connection widget.
Workflow Status	Overview of started, now open, stopped, and completed workflows since the time specified, including statistics on completion lead time. This widget replaces the ION Workflows widget.
Task Status	Overview of these tasks: <ul style="list-style-type: none"> <li>• Tasks that are now open, including new, unassigned, and assigned tasks</li> <li>• Generated tasks</li> <li>• Canceled tasks</li> <li>• Tasks that were completed since the time specified</li> </ul> This widget replaces the ION Tasks widget.

Name	Description
Alert Status	<p>Overview of these alerts:</p> <ul style="list-style-type: none"> <li>Alerts that are now open, including new, unassigned, and assigned alerts</li> <li>Generated alerts</li> <li>Canceled alerts</li> <li>Alerts that were completed since the time specified</li> </ul> <p>This widget replaces the ION Alerts widget.</p>
Alert List	<p>Shows a list of all open alerts sent to the current user.</p> <p>These features are added to the Alert List widget:</p> <ul style="list-style-type: none"> <li>Alerts can have a due date.</li> <li>Alerts that are escalated related to a due date have the escalated label.</li> <li>The alert list is updated automatically based on notification events sent from the server; a maximum of 200 alerts can be displayed.</li> </ul>
Task List	<p>Shows a list of all open tasks sent to the current user.</p> <p>These features are added to the Task List widget:</p> <ul style="list-style-type: none"> <li>Tasks can have a due date.</li> <li>Tasks that are escalated related to a due date have the escalated label.</li> <li>Tasks details can be viewed in a custom form if this was configured in the workflow model.</li> <li>Users can choose to enable or disable viewing the workflow progress in the task details screen.</li> <li>The task list is updated automatically based on notification events sent from the server; a maximum of 200 tasks can be displayed.</li> </ul>
Notification List	<p>Shows a list of all open notifications sent to the current user.</p> <p>These features are added to the Notification List widget:</p> <ul style="list-style-type: none"> <li>This is a new widget that shows only notifications.</li> <li>Workflow attachments are displayed in notifications.</li> <li>The notification list is updated automatically based on notification events sent from the server; a maximum of 200 notifications can be displayed.</li> </ul>

Name	Description
Workflow Viewer	<p>This widget shows the workflow status diagram for a task from the Task List widget or for a notification from the Notification List widget. These features are added in the Workflow Viewer:</p> <ul style="list-style-type: none"> <li>• The sub-process elements from the workflow diagram are displayed and these can be expanded or collapsed.</li> <li>• A refresh button is available to refresh the status of the running workflow from the Workflow Viewer widget.</li> <li>• The workflow activity for the selected task or notification is highlighted in the diagram.</li> <li>• The user can enable the feature to highlight the path taken from the beginning of the workflow to the current task or notification.</li> </ul>
Start Workflows	<p>With this widget authorized users can view and manage workflows that are configured in ION Desk. You can start new workflows and follow their status and you can cancel workflows that you have started.</p>
Task Manager	<p>With this widget authorized users can view and manage tasks from workflows configured in ION Desk. You can view the tasks with any status. For tasks that are open, you can add more users to the distribution list and you can re-assign a task.</p>
Alarms	<p>With the Alarms widget users can create monitors based on the alarm templates defined in ION Desk. The functionality is similar to the Alarms mobile application. Users can define their own alarm conditions and a distribution list for the alert that is created when the condition is met.</p>
Consumption by connection points	<p>Counts the messages that are successfully delivered to destination connection points, excluding ION API, from the beginning of the specified time.</p>

See the *Infor Ming.le User Guide*.

## Chapter 2: ION Desk overview

ION Desk is a browser-based interface that you use to configure and monitor all Infor ION services.

ION Desk consists of a web page that can show multiple application pages.

You can switch the desk to your language by changing these settings:

- The language and locale of Infor OS to your language in the user settings.
- The language settings of the browser to your language.

The ION Desk runs in Infor Ming.le. The user's Common Name and the current tenant name are displayed in the upper right corner of Infor Ming.le.

For ION Desk, all resolutions of 1280x1024 and higher are supported. Collapsing the menu and the left-hand-side header panel provides an optimal user experience.

**Note:** Excessive zooming, more than 100%, is not supported.

In the upper part of the ION Desk, this hamburger icon is displayed:



Click this icon to open the ION Desk menu, which contains entries to the application pages.

This table shows the main areas of the menu:

Level 1	Description
Home	Status overview page. Shows the important counters for the active objects in ION.
Reports	Show reports. This menu entry is only available when reports are generated.
Connect	Area to model connection points, data flows, and so on. View statistics and historic data.
Monitors & Workflows	Area to model workflow definitions, event monitors, alarm templates, and so on. View statistics and historic data, such as monitors and workflows that are archived.
OneView	Troubleshooting tool to trace the data throughput in ION.
Data Catalog	Area to manage object schemas and other metadata.
Data Lake	This area provides management tools for Infor Data Lake, such as storage policies, restore, and statistics.
Configuration	Configuration area for miscellaneous configurations and downloads.

Level 1	Description
Authorizations	Configuration area for authorizations and page access.
About ION	Shows the dialog box with the version information.

### Menu hierarchy

This table shows the menu hierarchy:

Level 1 categories	Level 2 pages or categories	Level 3 pages
Home		
Reports		
Connect	Data Flows	
	Connection Points	
	Mappings	
	File Templates	
	Enterprise Locations (CE-only)	
	Active Data Flows	
	Active Connection Points	
	Active Routings	
	Error BODs	
Monitors & Workflows	Monitors	
	Alarm Templates	
	Workflows	
	Activation Policies	
	Workflow Schedules	
	Business Rules	
	Active Monitors	
	Active Alarm Templates	
	Active Workflows	
	Active Activation Policies	
	Active Workflow Schedules	
	Activities	
	Alarms	
	Archive	Archived Monitors
		Archived Alarm Templates



Level 1 categories	Level 2 pages or categories	Level 3 pages
		Archived Workflows
		Archived Activation Policies
		Archived Activities
OneView		
Data Catalog	Schema Extensions	
	Object Schemas	
Data Lake (CE only)	Data Lake Overview	
	Storage Policies	
	Restore	
Configuration	Codes	
	Error Reporting	
	ION Service	
	Downloads (CE-only)	
Authorizations	Desk Profiles	
	Desk Permissions	
	My Permissions	
	Workflow Authorizations	
About ION		

Most pages are on level two in the hierarchy. For example, to open the **File Templates** page, select **Connect > File Templates**.

Some pages are on level three in the hierarchy. For example, to open the **Archived Monitors** page, select **Monitors & Workflows > Archive > Archived Monitors**.

### Accessing the Reports page

If you have generated one or more reports from ION Desk, the hamburger icon gets a blue icon on top (“new” indicator). If you open the menu, a new entry, **Reports**, is displayed in the top of the menu. A counter next to **Reports** shows how many reports are generated. Select **Reports** to navigate to the **Reports Overview** page.

### Types of pages

There are two types of overview pages in the ION Desk: Tiles and Grids. The tiles are used for modeling objects. The grids are used for the status overview pages. The home page is also tiles-based. The tiles on the home page represent entries to other manage pages.

## Home page

Tiles are used for the overview of the active objects and errors currently registered in ION. Each tile shows the important status counters for each object. The counters are linked to the manage pages of the objects. You can receive the latest counters by moving the pointer over each tile and click **Refresh**.

This table describes each tile with its counters for each type of objects:

Name	Counter Type	Description
Data Flows	active	Opens a list of active data flows.
Connection Points	active, paused, in error	Opens a list of active connection points that is filtered by the counter type.
Error BODs	unhandled errors	Opens a list of error messages that is filtered by the counter type "Unhandled".
Event Monitors	active, paused	Opens a list of active monitors that is filtered by the counter type.
Alarm Templates	active, paused	Opens a list of alarm templates that is filtered by the counter type.
Workflows	active, running	Opens a list of active workflows.
Workflow Schedules	running, paused, completed	Opens a list of active workflow schedules.
Activation Policies	active, paused	Opens a list of activation policies that is filtered by the counter type.
Activities	open alerts, open tasks	Opens a list of activities that is filtered by the type.

## Model overview

Tiles are used for the modeling objects overview. Each tile shows the most important information about the object, such as status, name, description, version, history timestamp, warnings, and errors.

You can delete, duplicate, export, activate, and deactivate objects from their tiles without opening the modeler.

**Note:** When you duplicate an object, sensitive data such as password information is not duplicated.

On the tiles page, objects are sorted from the most recently created or updated. The first position is reserved for the **Add** tile, which is the only tile not representing an object but an action.

To start creating a new object definition, click the **Add** tile. To edit an existing object definition, click the tile representing it.








The number of available object definitions are shown on the button bar.

**Note:** The system can only retrieve to 10,000 items at a time. Use filtering to limit the results.

## Tile toolbar

Use the tile toolbar for object-specific actions such as delete, duplicate, and activate.






This table shows the actions that are available from each tile:






Name	Type	Description
Delete		Delete the item. The action is not applicable for the active (running) item.
Duplicate		Open a copy of the item in the modeler.
Export		Export the item to a file.
Activate		Activate the item, so it is used in the ION Service.
Deactivate		Deactivate the item, so it is no longer used in the ION Service. The action is not applicable for already inactive items.
Protect on Export		Marks the item to be protected on export.
Do not protect on Export		Marks the item to not protect on export.
Usage	Text button	Show the list of configurations that use the item.

## Single-select mode

You can only work with one object definition at a time by default. To select multiple objects, click **Select** in the top button bar.

This table describes the button bar in single-select mode:











Name	Type	Description
Tiles view		Show definitions as tiles.
Grid view		Show definitions as a grid.
Refresh		Refresh the data on the page.
Import		Import item(s) from a file.
Select		A switch button is used to enable multi-select mode for bulk actions, such as exporting several definitions. When clicked, the button bar is updated to show the actions applicable to the selected item(s).
Filter By Keyword	Text field	Filter the view to show only those items where name, description, or author (if applicable) contain the text typed into the field. The icon turns blue when the view is filtered.






Name	Type	Description
Filter By Type		Filter the view to show only those items, which type matches the selection. On click, a list of available types is shown. You can select multiple items and the view is automatically updated to match your selection. The icon turns blue when the view is filtered. To close the context menu, click the filter icon again or click elsewhere.
Filter By Active	 	All active items are displayed when green. On click, active items are removed from the view. The icon becomes grey.
Filter By Inactive	 	All inactive items are displayed when red. On click, inactive items are removed from the view. The icon becomes grey.

## Multi-select mode

To work with multiple object definitions, click **Select** in the button bar.

This table describes the Button bar in multi-select mode:



Name	Type	Description
Tiles view		Show definitions as tiles.
Grid view		Show definitions as a grid.
Delete		Delete the selected item(s). Not applicable for the active (running) items.
Export		Export the selected item(s) to a file.
Activate		Activate the selected item(s), so it is used in the ION Service.
Deactivate		Deactivate the selected item(s), it is no longer used in the ION Service. Not applicable for already inactive items.
Cancel Select		A switch button is used to exit multi-select mode. When clicked, the button bar changes according to the table above.
Select All		Select all items in the tile view that match the used filters.
Protect on Export		Mark the item to be protected on export.
Do not protect on export		Marks the item to not protect on export.

Name	Type	Description
Filter By Keyword	Text field	Filter the view to show only those items where name, description, or author (if applicable) contain the text typed into the field. The icon turns blue when the view is filtered.
Filter By Type		Filter the view to show only those items, which type matches the selection. On click, a list of available types is shown. You can select multiple items and the view is automatically updated to match your selection. The icon turns blue when the view is filtered. To close the context menu, click the filter icon again or click elsewhere.
Filter By Active	 	All active items are displayed when green. On click, active items are removed from the view. The icon becomes grey.
Filter By Inactive	 	All inactive items are displayed when red. On click, inactive items are removed from the view. The icon becomes grey.

Alternatively, you can perform actions per object from the tile toolbar. The tile toolbar is displayed when the user moves the pointer over the tile on the overview page.

## Grids overview

This table shows the basic buttons on the grid-based overview pages:










Name	Type	Description
Back		Navigate back.
Refresh		Refresh the data on the page.

Text buttons are available depending on the context, for example: **Resume**, **Pause**, **Alerts**, **Triggers**, **Model**. For more details, see the correct sections of this user guide.

Actions applicable to the selected items in a grid are shown on the action panel sliding out on top of the grid. This panel is hidden until selection is done. Some actions are applicable to one item only. For example, the text buttons "details" or "model" are disabled when multiple items are selected.

## Details page

This table shows the button bar of the details pages:

Name	Type	Description
Back		Navigate back.
Save		Save the item. Save is applicable for model pages only.
Refresh		Refresh the data from the runtime. Refresh is applicable for manage pages only.
Duplicate		Open a copy of each selected item in the modeler.
Export		Export the item to a file.
Activate		Activate the item, so it is used in the ION Service.
Deactivate		Deactivate the item, so it is no longer used in the ION Service. The button is not applicable for already inactive items.
Protect on Export		Marks the item to be protected on export.
Do not protect on export		Marks the item to not protect on export.
Usage	Text button	Show the list of configurations that use the selected item.

Text buttons are available depending on the context, for example: **Resume**, **Pause**, **Alerts**, **Triggers**, **Model**. For more details, see the proper sections of this user guide.




## Indicators


The places where you can find additional information about the object definitions are:

- Next to the status badge on the object tiles.
- In the overview grids in a separate column.

The provided information can contain a change, an error indicator or an approval status applicable to that object.

This table shows the indicators:

Name	Type	Description
Change indicator		Indicates the object definition is changed since the last activation.
Error indicator		Indicates the inactive object definition contains errors
Protected on Export		Indicates the object that can be protected on export.

Name	Type	Description
Protected		Indicates the object is protected.

## Bookmarking pages

You can bookmark main pages and details pages in ION Desk. To bookmark a page, click the **Bookmarks** icon in the Infor Ming.le top navigation panel. For details on how to use bookmarks, see the *Infor Ming.le User Guide*.

For bookmarking your search in OneView, see [Search documents in ION OneView](#) on page 362.

## Starting ION Desk

To start ION Desk:

- 1 Navigate to Infor Ming.le.
- 2 Click the Application Switcher icon to open the Application Switcher panel.
- 3 Click the ION Desk icon:



ION Desk starts inside the Infor Ming.le window, using the same log in credentials as you used in Infor Ming.le. A loading progress status bar is displayed in the upper-right corner.

In ION Desk, you can see only models that belong to the current tenant. You can see models created by all users from the same tenant.

### Note:

- ION Desk can be used simultaneously by multiple users, as long as they do not update the same data simultaneously.  
If users change different objects, such as different connection points or different workflow definitions, these changes are saved independently.  
If several users simultaneously change the same model, only the changes of the second user who saves are successful; the changes of the other user are discarded without an error message.
- For an optimal view of the ION Desk, we recommend that you use a resolution of at least 1280 x 1024 pixels. For lower resolutions, you can use the scrollbars or close some panels.

## Viewing ION Desk version details

To view ION Desk version details:

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **About ION**.
- 3 The ION Desk About dialog box is opened. This information is displayed:

**The Infor ION Cloud Edition version**

The ION CE version, including the patch number.

**Copyright notice**



---

## Chapter 3: ION Connect concepts

The main purpose of ION Connect is to deliver business documents from one place to another.

ION uses these items:

- Documents.
- Connection points that can send or retrieve documents.
- Document flows that can be modeled to deliver the documents per your specific business processes.
- Data Lake Flows, special type of data flows, that can be modeled to ingest the documents to Infor Data Lake or retrieve documents from Infor Data Lake.

**Note:** Avoid using messages that are larger than 5 MB. ION can handle larger files but that depends on the available system resources for ION. In that case, a correct operation of all ION functionality cannot be guaranteed in all circumstances.

## Documents

In ION Connect these types of documents exist:

- Business Object Documents (BODs)  
BOD is a standard format defined by Open Application Group ([www.oagi.org](http://www.oagi.org)).  
Standard Infor BODs are available out of the box in the registry that is delivered with ION. You can define your own BODs in the registry as well.  
For a basic description about the BOD structure see "Business Object Document (BOD)" later in this topic.  
For more specific information, see the *Infor ION Development Guide*
- JSON documents  
JSON (JavaScript Object Notation) is an open-standard lightweight data-interchange file format.  
No standard JSON documents are delivered with ION. You can import a custom JSON document schema to the registry. You can automatically generate a JSON document schema and add it to the registry in some connection points.
- DSV documents  
DSV (delimiter-separated values) documents store tabular data in plain text. Each row in a delimited file contains one data record. Each record contains one or more fields, that are separated by a delimiter. The delimiter can be any character, but most commonly used are comma or tab.

No standard DSV documents are delivered with ION. You can import a custom DSV document schema to the registry.

- ANY documents.

ANY document is defined just by its name. Content structure is not defined or unknown. Some examples for usage of ANY documents are: CSV files, unstructured text, xls, pdf, jpeg, etc.

### Business Object Document (BOD)

A BOD (business object document) is a message that is sent from an application to one or more other applications. This type of message informs an application of a change to a business object that took place in another application, or requests an application to update a business object.

A BOD contains two parts: a noun and a verb.

#### Nouns

A noun is a set of business data contained in a BOD. The noun represents the properties of one business object. Examples of nouns are SalesOrder, Item, and BusinessPartner. In ION Desk, a noun is called an application document.

A BOD definition contains one noun definition. A message can contain multiple instances of the same noun definition. For example, a BOD can contain multiple SalesOrders, but it cannot contain SalesOrders and PurchaseOrders at the same time. However, usually a BOD will only contain one noun instance.

#### Verbs

The verb describes the action that is requested for the noun. This table shows the request verbs supported by Infor:

Request verb	Description
Sync	A synchronization message containing changes that took place to a business object. A Sync message is sent by the owner of the data and can be delivered to any other application for which this information is relevant.
Process	A request to create a business object or to apply changes to an existing business object. A Process message is sent from any application to the application that owns the data. The owner will send an Acknowledge message in response to the Process request. The loaded document can be refused.
Get	A request to get the details for a business object. A Get message is sent to the owner of the data. The owner will send a Show message in response to the Get request.
Load	The Load verb is used when a document is created by an application that will not be the owner. A Load message is sent to the owner of the data. The loaded document cannot be refused.
Post	<b>Note:</b> This verb is deprecated but still supported for compatibility. The Post verb is similar to the Process verb, but it does not trigger the creation of an Acknowledge message.

Request verb	Description
Update	The Update verb is used when data is changed by an application that does not own the data. The Update verb is similar to the Load verb, in the sense that it must be accepted by the owner. Namely, the Update message informs the owner of the data that an event took place and what data was changed by the event.

Verbs are also used for response messages. The response verbs supported by Infor are shown in this table:

Response verb	Description
Acknowledge	An Acknowledge response is sent in reply to a Process request. An Acknowledge response indicates whether the object to be processed was accepted, modified, or rejected.
Show	A Show response is sent in reply to a Get request. A Show message can also be sent without a preceding Get request. For example, an application can send Show messages to load new applications with initial data
Confirm	A Confirm verb is used when a failure occurs. A Confirm verb is processed within ION and is not routed to any other application. The Confirm verb is used only for the BOD noun. The ConfirmBOD contains a copy of the original message, to enable an ION administrator to resubmit the same message after fixing the cause of the problem.

## Messages

In addition to the actual document content, the messages handled by ION contain additional technical details, such as the address of the sender. Where required, additional details are provided later in this guide.

## Connection points

ION connects to a connection point, to retrieve or send documents, for example other Infor Applications or database files through Enterprise Connector.

## Data flows

The most important items in ION Connect are the data flows.

A data flow is a sequence of activities that send or receive documents. Data flows are event-driven. When a document is published by an application, the next step in the flow is triggered. Each flow starts and ends with one or more connection points. Connection points can be reused in multiple data flows, and the same connection point can be used multiple times in a data flow.

---

## Document flows

Usually, a document flow represents a business process.

For example:

- An invoicing process where an invoice is created for a delivery that was shipped.
- A maintenance process where a service order is planned based on a customer repair request.

In a "simple" document flow, two connection points are involved. One connection point sends a specific type of documents, and the other connection point receives those documents. Document flows can be more complex.

In addition to connection points, document flows can contain other items:

- Mappings are used to translate a document to another format. For example, if you retrieve custom sales order documents from your database, you must translate them to standard Infor BODs to load them into an Infor application.
- Scripts are used for executing custom Python code on incoming documents. For example, scripts can be used to facilitate and solve common use cases such as data object format conversion, data mapping, complex calculations and transformations.
- Parallel flows are used if a document must be sent to multiple connection points. Or if documents from multiple connection points must be sent to a single connection point.
- Filters are used to limit the number of messages that are delivered to a connection point. For example, if only documents having a specific status are relevant for an application, you can filter out the documents having another status.
- Content-based routings are used to send a document to a subset of available destinations based on the document contents. For example, if three warehouses exist, each having their own warehouse management system, a `SalesOrder` message can be routed to one of these warehouses. The warehouse code that is specified in the `SalesOrder` message is used. The execution of document flows is fully automatic. After a flow is defined and activated, the relevant documents are routed accordingly.

## Data Lake flows

A Data Lake flow is a sequence of activities that results in sending data into Data Lake or sequence of activities starting with retrieval of data from Data Lake.

In a "simple" Data Lake flow, two connection points are involved. One Data Lake connection point sends a specific type of documents, and the other connection point receives those documents. Or a connection point sends a specific type of documents, and the Data Lake connection point receives those documents. Data Lake flows can be more complex.

Some activities are not allowed to be modeled in Data Lake Flow.

See [Creating and using data flows](#) on page 43.

Asynchronous activities are not allowed to be modeled in the middle of a Data Lake flow. Specifically these connection points:

- Application (IMS)

- Application (in-box/outbox)
- LN
- CRM Business Extension
- File
- ION API (Send, Read)
- Database (Read, Send, Request/Reply)
- Message Queue (JMS)

Documents that are modeled in Data Lake flows are subject to storage policies.

See [Storage policies](#) on page 483.

In addition to connection points, Data Lake flow can contain other items such as:

- Mappings that are used to translate a document to another format.
- Scripts that are used for executing custom Python code with a document input.
- Parallel flows that are used when sending documents from multiple connection points to Data Lake.

## Compare data flows and workflows

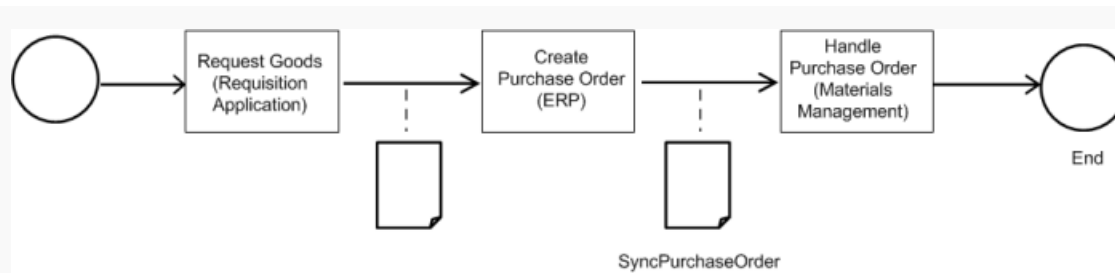
In ION, workflows can be defined and executed. Data flows and workflows use a comparable diagramming technique. Some significant differences exist between them. The main differences are listed in this table.

Workflows	Data flows
Focus is on user tasks. Tasks are distributed to users through Infor Ming.le.	Focus is on routing of documents. Applications are integrated automatically without user intervention
The process is managed by ION. The workflow controls which tasks are created and monitors the task status.	ION only delivers the documents. The data flow is event-driven; ION is passive, waiting for the next event to be published by an application. The ION data flow does not trigger events or actions in applications.
The amount of data that is handled in the workflow is limited to the workflow parameters and structures.	The documents to be delivered can be complex.
Applications can trigger a workflow by sending a Process. Workflow BOD can receive an Acknowledge Workflow BOD.	Applications communicate with data flow using documents.

For more information about modeling workflows that involve user tasks, see [Workflow](#) on page 246.

### 1. Business flow to handle requisitions

This diagram shows a flow to handle requisitions. A request for some goods is translated to a purchase order that is handled by a materials management application.

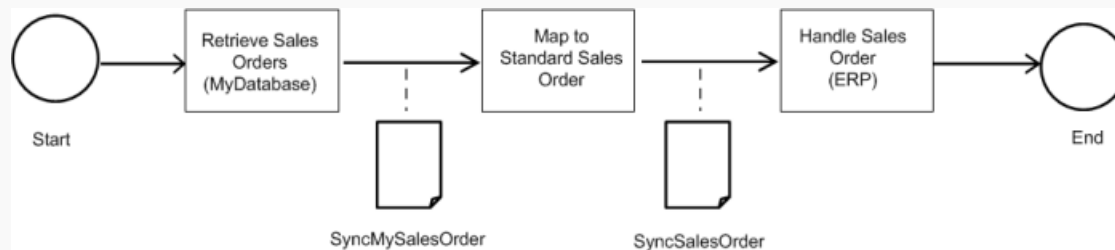


When the requisition application publishes a `ProcessRequisition`, ION delivers it to the ERP application. Because ION Connect is event-driven, it does not force the ERP system to create a purchase order. It delivers the request for a new requisition and then waits for the next event to happen. When the ERP system publishes a `SyncPurchaseOrder`, that document is delivered to the MaterialsManagement application.

ION automatically handles reply BODs. In this case, when the ERP system sends out an `AcknowledgeRequisition` in reply to the `ProcessRequisition`, that document is delivered to the Requisition application.

## 2. Business flow to load orders

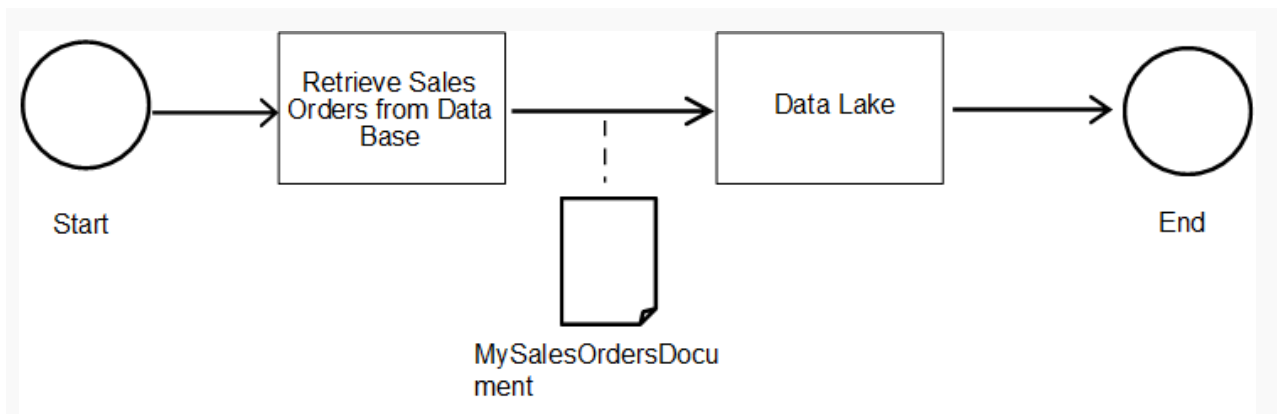
The diagram shows a flow to retrieve new or changed sales orders from a database and send them to an Infor ERP application. If the structure of the sales orders in the database differs from the standard `SalesOrder` BOD, a mapping activity is added to translate the custom sales order data into a standard `SalesOrder` BOD.



ION is event-driven and a database connection point is passive. Therefore, a schedule is used in ION to regularly read the data from the database.

## 3. Data Lake Flow to load historical Sales Orders

The diagram shows a flow to ingest the historical Sales Order into Data Lake.



After the message is delivered to ION, the message is sent into Data Lake.

## Message Routing

Documents can be routed in data flows either implicitly and explicitly.

With implicit routing, or routing by to logical ID, a `to logical ID` header is used to find an appropriate endpoint. The header is sent together with the message. The implicit routing is mainly used together with Request-Reply BOD messaging pattern.

The implicit data flow:

- 1 You model a data flow from Application A to Application B.
- 2 Application A sends a request message, `Process` or `Get`, to application B.
- 3 Application B generates a reply document, `Acknowledge` or `Show`. The logical ID from the `From Logical ID` request message is added to the `to logical ID` in the reply message.
- 4 The reply message is received by ION. The `to logical ID` is read and the message is routed directly to the Application A.

Explicit routing, or routing by data flow, uses the data flow model to find the endpoint. Explicit routing of reply messages means that you can control the flow of reply messages in the same way as for request messages.

The explicit data flow:

- 1 You model a data flow from application A to application B.
- 2 You model a data flow from application B to application A.
- 3 Application A sends a request message, `Process` or `Get`, to application B.
- 4 Application B generates a reply document, `Acknowledge` or `Show`. The logical ID from `to logical ID` request message is added to the `to logical ID` in the reply message.
- 5 The reply message is received by ION and the relevant data flow is found.
- 6 The message is routed to application A according to data flow for reply message, `to logical ID` is ignored.

Explicit routing of reply messages can be used to control reply messages. In situations where the `to logical ID` cannot be used to route the message to the correct target. Explicit routing has priority over implicit routing. If a message with `to logical ID` is received and Data Flow is found, the Data Flow route is used instead of `to logical ID`.

**Note:** When a data flow is used to route reply messages, that is also containing valid `to logical ID`, there are two ways to route. When data flow is active, `to logical ID` is ignored and routing is done according to data flow. When data flow gets deactivated, routing according to data flow is not possible. The message is delivered with implicit routing according to `to logical ID` possibly to different connection points.

## Examples for using explicit routing on reply messages

This section describes examples for these use cases:

- Inverse Mapping
- Filter-out reply
- Different endpoint for reply
- Generate reply

### Inverse Mapping

Consider this data flow for request document:

```
A --> Process.ItemMaster --> Mapper --> Process.MyItem --> B
```

Application A sends a `Process.ItemMaster`. The mapper changes the noun to a `Process.MyItem` and delivers it to application B.

- Implicit routing  
Application B sends an `Acknowledge.MyItem` to application A. Application A is expecting `Acknowledge.ItemMaster` instead of `Acknowledge.MyItem`.

- Explicit routing  
Additional data flow for reply can be modeled:  

```
B --> Acknowledge.MyItem --> Mapper --> Acknowledge.ItemMaster --> A
```

Application B sends an `Acknowledge.MyItem` to Mapper. It is converted to the `Acknowledge.ItemMaster` and sent to application A.

### Filter-out reply

Consider this data flow for a request document:

```
A --> Sync.SalesOrder --> Mapper --> Process.SalesOrder --> B
```



Application A sends a `Sync.SalesOrder`. The mapper changes the verb to a `Process.SalesOrder` and delivers the verb to application B.

- Implicit routing  
Application B sends an `Acknowledge.SalesOrder` to application A. Application A is not expecting `Acknowledge.SalesOrder` since it sent a `Sync` verb and generates a `Confirm.BOD`.
- Explicit routing  
You can model an additional data flow for reply:  
`B --> Acknowledge.SalesOrder --> Filter`  
Application B sends an `Acknowledge.SalesOrder` to the Filter component. All reply documents are filtered out and no `Confirm.BOD` is generated.

## Different endpoint for reply

Consider this data flow for request document:

```
A --> Process.SalesOrder --> B
```

Application A sends a `Process.SalesOrder` to application B:

- Implicit routing  
Application B sends an `Acknowledge.SalesOrder` to application A. Application A can or cannot handle the response.
- Explicit routing  
You can model an additional data flow for reply:  
`B --> Acknowledge.SalesOrder --> C`  
Application B sends `Acknowledge.SalesOrder` to Application C. Application C accepts the acknowledge.

## XML document from external source

When integrating with third party applications, ION converts the data from the external source to a custom document. However when the external application provides XML data as input to ION it must be in a valid XML format. Characters in the XML content such as `&`, `>`, `<`, single quote and double quote must be formatted using appropriate standard XML escape characters.

## Chapter 4: ION Connect Modeling

When modeling in ION Connect you can use two approaches.

For example:

- Create a data flow and define the details for the required connection points on the fly.
- Create the connection points and subsequently create a data flow using these connection points.

Connection points can be used in multiple data flows. Changing the properties of a connection point impacts all data flows that use that connection point.

For Infor applications, default connection point definitions can be available. Additionally, if you use multiple Infor applications, standard document flows can be available.

For more details on how to deploy connection points or document flows, see [ION Connect import and export](#) on page 191.

### Prerequisites

Before creating data flows, the metadata for documents to be used must be available in the ION Registry. Standard Infor documents are available by default. For custom documents, the definitions must be added. You can upload custom document definitions in ION Desk. Select **Data Catalog > Object Schemas**.

To create custom document definitions, see the *Infor ION Development Guide*.

In the rest of the document, ION Registry is referred to as "the Registry".

### Using the modeler

This table shows the areas in the modeler:

Area	Description
Summary pane	Contains the summary of the document flow including the notification panel for errors and warnings. The panel is on the left from the modeling canvas. This panel can be collapsed.

Area	Description
Toolbox	Contains the items that are used to create a model by dragging and dropping them onto the modeling canvas. The panel is above the modeling canvas. This panel can be collapsed.
Modeling canvas	Contains a graphical representation of the data flow. The representation of the flow on the modeling canvas is based on BPMN 2.0.
Properties pane	Contains the properties of the item that is currently selected in the modeling canvas. The panel is under the modeling canvas. This panel can be collapsed.

## Creating and using data flows

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Connect > Data Flows**. An overview of existing data flows is displayed.
- 3 Click **+Add** to open the modeler.  
A list of data flow types is displayed:
  - Document flow
  - Data Lake flow
- 4 Select one data flow type.  
The modeler page is displayed.
- 5 Specify a name for the data flow.
- 6 Model the flow by dragging items from the toolbox to the model line between the Start and the End points.
- 7 Specify the properties for each item in the model. Click the item to view or change the properties.
- 8 Right-click the item to delete, copy, or cut it.
- 9 Save the model.
- 10 Activate the model and navigate back to the overview page. Alternatively, you can activate the model later from its tile using the tile toolbar.  
Click **Remove connection points** to clear the connection points settings from the model. This is helpful when you must distribute template data flows although the connection points to be used are not known upfront.

## Modeler toolbox

The modeler toolbox contains these categories of items:

- Core activities - to send, receive documents.
- Data Lake activities - to ingest documents to or retrieve documents from Data Lake.

- Supporting activities - to set parameters.
- Flows - to route or filter documents.

**Note:** Only the subset of activities and flows are used in the Data Lake flow modeler toolbox.

## Overview


For consistency and to distinguish connection point activities from other activities in ION, the activities colors in the toolbox were updated. The colors relate to the activity type in the desk data flow and workflow modelers.









This list shows the colors that are used for which activities:




- **ORANGE** for core activities in:
  - Data flows: Connections, for example, connection points.
  - Workflows: User interaction, for example, task, notification etc.
- **RED** for Data Lake activities in:
  - Data Lake flows: Ingest, Retrieve
- **GREEN** for supporting activities in:
  - Document flow: Mapping, splitting, workflow.
  - Document flow: Mapping, scripting, splitting, workflow.
  - Data Lake flows: Mapping
  - Data Lake flows: Mapping, scripting
  - Workflow: System activities such as set parameter.
- **BLUE** for flow activities in:
  - Document flow: Filter, parallel, routing.
  - Data Lake flows: Parallel
  - Workflow: Filter, parallel, loop back, sub-process.
- ION API activity is green in the workflow modeler, similar to set parameter, exit point etc. In the workflow modeler, the core activities are user interactions such as tasks and notifications.
- ION API activity is orange in the data flow modeler, similar to other connection points. In the data flow modeler, the core activities are application interactions such as connection points.

## Data Flow Modeler - Activities

This table shows the toolbox icons for the Activities category:



Name	Widget	Description
Application		<p>Activity from an application, which sends or receives messages. For this activity, you must create a new application connection point or select an existing application connection point.</p> <p>In case of multiple application instances that have the same role in the business process, you can select multiple application connection points. For example, when you have multiple instances of a warehouse management system, one per warehouse, that must receive the same master data documents.</p>


Name	Widget	Description
Business		<p>Activity to a business application that sends or receives application-specific data and transforms it to BOD. For Example, a Business Activity of type SAP is used to communicate with SAP and transform SAP IDOC to and from BOD.</p> <p>For this activity, you must create a new Business connection point or select an existing Business connection point.</p>
Network		<p>Activity to send messages to or receive messages from the different tenant.</p> <p>To use this activity, you must be provisioned with a 'Network' connection point.</p> <p><b>Note:</b> Not available in the Data Lake flows and when no Network connection point is provisioned.</p>
Database		<p>Activity to read or write a database.</p> <p>For this activity, you must create a new database connection point or select an existing database connection point.</p> <p>See <i>Infor ION Technology Connectors Administration Guide</i></p>
File		<p>Activity to read or write text files.</p> <p>For this activity, you must create a new File connection point or select an existing File connection point.</p> <p>See <i>Infor ION Technology Connectors Administration Guide</i>.</p>
ION API		<p>Activity to Send to API, Get from API or Trigger API call.</p> <p>For this activity, you must create a new ION API connection point or select an existing ION API connection point.</p> <p>See <i>Infor ION Technology Connectors Administration Guide</i>.</p>
Message Queue		<p>Activity to read or write a message queue.</p> <p>For this activity, you must create a new message queue connection point or select an existing message queue connection point.</p> <p>See <i>Infor ION Technology Connectors Administration Guide</i>.</p>
Web Service		<p>Activity to use documents from a web service or send documents to a web service.</p> <p>For this activity, you must create a new web service connection point or select an existing web service connection point.</p> <p><b>Note:</b> Not available in the Data Lake flows</p> <p>See <i>Infor ION Technology Connectors Administration Guide</i>.</p>
Mapping		<p>Activity to map a document to another type of document, or to change the data in the document.</p> <p>For this activity, you must create a new mapping or select an existing mapping. See <a href="#">Document mappings</a> on page 92.</p>

Name	Widget	Description
Scripting		<p>Activity to run scripts with custom Python code on incoming documents.</p> <p>For this activity, you must create a new script or select an existing script.</p> <p>See "ION Scripting".</p>
Workflow		<p>Activity to start a workflow. In this activity you select a workflow and link the input and output parameters of that workflow to attributes of the incoming document.</p> <p>When a document arrives at this activity, a workflow is started. When the workflow is completed, the output of the workflow is added to the document and the document flow continues.</p> <p><b>Note:</b> Not available in the Data Lake flows</p>
Splitter		<p>Activity to split one document containing multiple instances of the same object into separate documents. Each document contains one object together with a header and footer</p> <p><b>Note:</b> Not available in the Data Lake flows</p>

#### Data Flow Modeler - Flows

This table shows the toolbox icons for the Flows category:

Name	Widget	Description
Parallel		<p>Flow to use multiple activities in parallel. Use a parallel flow in these situations:</p> <ul style="list-style-type: none"> <li>• A document is sent to multiple connection points.</li> <li>• Documents are sent from multiple connection points to a single connection point.</li> </ul> <p>Do not define properties for a parallel flow. You can add or remove branches by right-clicking the diamond shape.</p>
Filter		<p>Flow to limit the number of messages that are delivered to a connection point.</p> <p>To define a filter, select one or more properties from the involved document and build a condition using these properties.</p> <p>See <a href="#">Filtering and content-based routing</a> on page 101.</p> <p>A filter can also be used if the previous activity provides multiple document types. In that case a filter is defined per document type. Not all document types require filtering.</p> <p><b>Note:</b> Not available in the Data Lake flows</p>

Name	Widget	Description
Routing		<p>Flow to send a document to a subset of available destinations based on the document contents. To define the routing, select one or more properties from the involved document and build conditions using these properties.</p> <p>For each branch a condition must be defined.</p> <p>See <a href="#">Filtering and content-based routing</a> on page 101.</p> <p>You can add or remove branches by right-clicking the diamond shape. A routing flow can be used if the previous activity only provides a single document type.</p> <p><b>Note:</b> Not available in the Data Lake flows</p>

## Using the notifications pane

The notifications pane contains errors or warnings for the current document flow and the used mappings, connection points, etc.

For example, an error is reported if a document is selected that cannot be received by the next activity.

Errors must be solved before activating the flow. For warnings you can choose whether they must be solved.

To navigate to the involved item in the modeling canvas, click the message line.

## Activating and deactivating data flows

A new or changed data flow is used in the ION Service after it is activated. You can save data flows or connection points without impacting the current ION Service behavior.

Activating and deactivating is done on the **Data Flows** page.

### Activating a data flow

- 1 Select **Connect > Data flows**.

A list of existing data flows is displayed.

- 2 Activate one data flow from the tile toolbar or enable the multi-select mode and select several data flows.

- 3 Click **Activate**.

Alternatively, click the data flow tile to open the modeler. Activate the model by clicking **Activate** in the modeler button bar.

Inactive flows can be activated, but also active flows can be reactivated.

If a connection point is used in multiple data flows, you can select them together and activate them all at once. If a data flow uses a new application or message queue connection point, ION starts reading messages from the Outbox or JMS message queue from the moment a data flow is activated. If you first activate one data flow and later another data flow, ION can already have processed documents from the outbox or message queue that were relevant for the second data flow, before that second data flow was activated.

Selecting a large number of data flows and activating them all at once, can cause a time-out error. In this case, reduce the selected data flows and try again. Group all data flows of a modified connection point together for the reasons explained earlier.

During activating, errors or warnings can be reported. Normally, an activation error is reported if one or more errors exist in the messages pane of the data flow modeler.

## Deactivating a data flow

- 1 Select **Connect > Data Flows**.

Existing data flows are displayed as tiles.

- 2 Deactivate an active data flow from the tile toolbar or enable the multi-select mode and select several data flows.

- 3 Click **Deactivate**.

Alternatively, click the data flow tile. The modeler is opened. Deactivate the model by clicking **Deactivate** in the modeler button bar.

## Changing data flows

- 1 Select **Connect > Data Flows**.

Existing data flows are displayed as tiles.

- 2 Click the data flow tile. The modeler is opened.

- 3 Change the model.

For details, see [Using the modeler](#) on page 42.

- 4 Save the model by clicking the **Save** icon.

- 5 Activate the model by clicking the **Activate** icon.

- 6 Return to the overview page.

You can also activate the flow from its tile toolbar.

If the data flow was already active, it is marked as 'changed since last activation'. ION Service is not changed until activating the changed data flow. When reactivating a data flow, the used connection points and mappings are also activated. If the connection points are used in another active data flow, the changes are also deployed in that data flow. For mappings the changes are applied to the data flow that is reactivated.



---

## Using multiple data flows

When using multiple data flows, the delivery of messages is optimized to some extent.

For example; two data flows are activated specifying that `SyncSalesOrder` must be sent from A to B. Then the `SyncSalesOrder` messages from A are delivered to B only once.

When in one of the data flows intermediate steps are defined such as mapping, filter or routing, then the same document can be delivered twice.

For example; the first data flow can have a filter between A and B. Only sales orders with status is **Open** are sent. The second data flow has a filter and only sales orders for accounting entity **A** are sent. In that case, if a `SalesOrder` document from A satisfies both filters, it is delivered twice to B.

How this is handled depends on the connection point. Infor Application connection points can avoid storing duplicate messages in the in-box. Even if a database, message queue, web service or file connection point has no state and the message is delivered twice.

The same document and sender application can be reused in both document flow and Data Lake flow. In this scenario, a document is sent to the subscriber that is specified in document flow and to the Data Lake.

For example; a document flow and Data Lake flow are activated specifying that `SyncSalesOrder` must be sent from A to B in document flow and from A to Data Lake in Data Lake flow. Then the `SyncSalesOrder` messages from A are delivered to B and to Data Lake.

## Creating and using API flows

The API Flows are used for ION API Orchestration. You can define synchronous business flows in which you can chain more APIs, execute conditional logic and expose the whole flow as a new API.

For example, you can model an Inventory check flow or a Price check flow. You can evaluate conditions in the request and invoke different ION APIs to get the data. You can enrich the response by adding more data from additional ION APIs. The final response can be presented back in the client preferred format. Only JSON request body and JSON response are supported by ION API Flows.

### Prerequisites

To create API Flows, you must have modeling authorization in the ION Desk. You must have the security role settings `IONDeskAdmin` and `MingleIONenabled`.

## Using the API Flow modeler

The API Flows modeler is comparable to other ION modelers, such as data flow and workflow.

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Connect > API Flows**. An overview of existing API flows is displayed.

- 3 Click **+Add** to open the modeler.
- 4 Specify a name for the API flow.
- 5 Specify the API Flow properties.  
See [API Flow Properties](#) on page 52.
- 6 Model the API calls by dragging items from the toolbox to the model line between the Start and the Endpoints.  
See [API Flow elements](#) on page 51.

## API Flow parameters

API flow parameters are used to move information in the API Flows.

All API Flow request parameters, ION API step output parameters and Decision output parameters are available in API Flow as API Flow parameters.

The flow parameters can be data types such as string or integer, and JSON object.

The API flow parameters can be referenced in input fields as `${<activityName>.<parameter Name>}`, for example `${getInventory.list}`. API Flow parameters coming directly from API Flow request are referenced using "input" instead of activity name, for example as `${input.store}`.

You can also use the help function to select an API Flow parameter. On the selected input field, press CTRL+SPACE. A list of available API Flow parameters is displayed. Select an API Flow parameter to add to the input field.

In case an API Flow parameter is a JSON object. You can specify elements from the parameters with JSON Path expressions. For example, if an API Flow parameter contains a JSON Object:

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
  },
}
```

```
"expensive": 10
}
```

The category value from the second book is required. You can use this JSON Path:

```
${getInventory.list}.store.book[1].category
```

The result is "fiction".

For more information about the JSON path syntax, see <https://github.com/json-path/JsonPath>

## Publishing

You can activate a flow directly from the modeler.

- 1 Save the flow.
- 2 Click **Activate**.

After saving and activating:

- The API documentation as per OPEN API 2 specification is generated.
- The API flow is published to the default suite on the ION API gateway.

The default properties are:


- Suite Name: Infor API flows
- Endpoint: The URL that consists of the API flow suite base URL and the endpoint and resource path of the API flow.



If the API flow must be accessed through a different suite, you can define a new proxy end point in that specific suite. The target must point to your API flow backend URL.

The special suite for API Flows in the ION API gateway is displayed after the activation of the first API flow. When you deactivate the last flow, the "Infor API Flows" suite disappears.

## API Flow elements

This table shows the API Flow definition and its predefined types of steps:

Step	Description
ION API 	A step to perform a call to an ION API operation. In this step, you can define an API request and output parameters. See <a href="#">ION API step</a> on page 54.

Step	Description
Parallel 	An unconditional execution of two or more execution branches in parallel. In this step, you can define parallel calls to multiple APIs. The API flow waits until all the API calls in each of the branches are completed. See <a href="#">Parallel API flows</a> on page 56.
Decision 	An evaluation of an expression that has several possible follow-up branches. The default branch is executed when values in the other branches do not match the expression result. In this step, you can define conditional flows based on API response code, response message etc. The decision step is exclusive. Only one branch is executed to avoid conflicts and merging of the results. See <a href="#">Decision API flows</a> on page 56.

## API Flow Properties

To get to the API Flow properties, click either the start or end events, or click the **Show API Flow properties** link on the left hand side panel.

### Request tab

Use the **Request** tab to specify the API operation, Method and URL, and Request parameters.

This table shows the properties of the API operation:

Name	Description
Method	Select an operation: GET, POST, PUT or DELETE.
Endpoint Name	Specify the API endpoint name, group of APIs in one API suite. In the first release this must be unique.
Endpoint Path	Specify part of URL path identifying the API endpoint. In the first release this must be unique.
Resource Name	Specify the API name.
Resource Path	You must specify the end of an URL path identifying the resource with the definition of all path parameters. Path parameters must be defined in curly brackets {}, for example: <code>/resourcepath/{PathParameterName}/</code>

Click **+** to add a new request parameter.

This table shows properties of request parameters:

Name	Description
Name	Specify the request parameter name. The names must be unique.
Description	Optionally, specify the request parameter description.
Parameter Type	Select the request parameter type: query, path or header. Path parameters are used to construct the URL path and cannot be blank.
Data type	Select the request parameter data type: string, number, integer, or boolean.
Required	Select, if the request parameter is required.

## Request Body tab

This tab contains the properties of the request body parameter.

This table shows the properties of the **Request Body** tab:

Name	Description
Presence	Select the presence of the request body: Required, Optional or None.
Content Type	The request content type is automatically defined as Application/JSON.
Name	Specify the request body parameter name.
Description	Optionally, specify the body parameter description.
Schema	Specify the request body schema, it must be valid a JSON object.
Example	Optionally, specify an example of a request body, it must be valid JSON.

## Response tab

This table shows the properties of the **Response** tab:

Name	Description
Content Type	Response content type is automatically defined as Application/JSON.

Name	Description
Body	<p>Specify response body, it must be valid JSON. You can use API Flow parameters with JSON Path expressions. See <a href="#">API Flow parameters</a> on page 50.</p> <p>Because the response must be valid JSON object, you cannot define the API Flow parameter alone, for example: <code>\${getInventory.list}</code>.</p> <p>If that is required, leave the response body field empty. The API Flow response defaults to the response of the previous ION API step.</p> <p>You cannot leave the response body field empty in case the last step is parallel branch.</p>

## Test tab

Use the **Test** tab to specify the request parameters and request body. Click **Test** to test the complete API Flow. The final API Flow response is displayed in the Response area.

**Note:** For testing the API flow is temporary published. The test request is using real API call, data can be changed by this action.

## ION API step

- 1 Specify a ION API step name in the **Name** field.
- 2 Specify a ION API step description in the Description field.
- 3 Click **SELECT**.

A dialog box is displayed to search for API operations which are available in ION API gateway.

- a Select a product from the drop-down list.
- b Specify a search string and click the **Search** icon.
- c Click **Show More**, if the operation you are searching for is not in the list.
- d Select an operation.
- e Click **OK**.

The product and operation description are displayed.

## Request parameters tab

This tab is contains the list of request parameters that are required to make the API call. This table shows the columns and description of the parameters:

Column	Description
Name	The request parameter name as defined in the API metadata. An asterisk indicates whether it is required to pass a value for this parameter.
Description	The request parameter description as defined in the API metadata.
Data Type	The request parameter data type as defined in API metadata.
Value	Specify request parameter value. It can be a constant value or you can use API Flow parameters. When API Flow parameter is a JSON Object, JSON Path expression can be used. See <a href="#">API Flow parameters</a> on page 50.

## Request Body tab

This tab contains the properties of the request body parameter. If the selected operation does not have a request body parameter, the controls on this tab are disabled.

This table shows properties of **Request Body** tab:

Column	Description
Content Type	If the request content type defined in API metadata is Application/JSON, it is automatically defined as Application/JSON.
Name	The name of the parameter of type body that was defined in the API metadata.
Description	The description of the parameter of type body that was defined in the API metadata.
Body	Specify request body, it must be valid JSON. You can use API Flow parameters. When API Flow parameter is a JSON Object, JSON Path expression can be used. See <a href="#">API Flow parameters</a> on page 50. If the request body is required for this operation, you cannot leave this text box blank. If the model schema is described in the API metadata, you can generate an example body. To specify the request body with a sample, click <b>EXAMPLE</b> .
Do not send request body	If the request body is optional, you can select this check box. A request is sent with a blank body.

---

## Output Parameters tab

On this tab you can define how API response is mapped to API Flow parameters. If the response content type defined in API metadata is Application/JSON, it is automatically defined as Application/JSON.

- 1 Click **+** to add new API Flow parameter.
- 2 Specify API Flow parameter name.
- 3 Select one of these Response Types:

### **BODY**

API Response Body. Can be selected only once.

### **CODE**

HTTP Response status code. Can be selected only once.

### **HEADER**

Response header name. The API Flow parameter name must match with the response header name.

## Test tab

On the **Test** tab you can specify API Flow Parameters that are used in request tabs as Input parameters of this ION API step.

Click **Test** to test the API step. The resulting API Flow parameters are displayed in the output parameters section.

**Note:** The test request is using real API call, data can be changed by this action.

## Parallel API flows

With the parallel flow, the modeler can execute multiple steps in parallel.

Each step can use the ION API Flow parameters from the previous steps as input. The parallel flow is expected to be executed faster than a sequential flow. The flow continues to the next steps only when all the branches of a parallel flow are completed.

The parallel block does not have properties. You can add or remove branches by right-clicking the diamond shape

## Decision API flows

With the decision steps, the modeler can decide which branch (steps) must be taken depending on values of input API Flow parameters.



You can add or remove branches by right-clicking the diamond shape. There is always one default branch. It is the one at the bottom and marked in the model with the small line across the input line. Default branch cannot be removed.

The flow only executes the first, from top, matching branch to avoid conflicts and results being merged.

API Flow parameters that are created in the steps inside a decision flow cannot be referenced directly. You must use a decision step name as an identifier instead of an individual step name.

## Properties tab

Specify a decision step name in the **Name** field.

Specify a decision step description in the **Description** field.

## Decision Parameters tab

- 1 Click **+** to add new decision parameter.
- 2 Specify a decision parameter name. The names must be unique.
- 3 Specify a decision parameter value.

It can be a constant value or you can use API Flow parameters. When API Flow parameter is a JSON Object, JSON Path expression can be used.

See [API Flow parameters](#) on page 50.

## Conditions tab

On the Conditions tab, you must specify two parts of the definition:

- 1 Expression

Specify the expression that is used to decide which branch must be executed. Define the expression with Javascript (nashorn).

Decision parameters are referenced as, `$.<decisionParameterName>` for example `$.book`.

In case the Decision parameter is a JSON object. You can specify elements from a parameter with JSON Path expressions. For example if the Decision parameter contains JSON Object:

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      {
```

```

        "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
    },
    "bicycle": {
        "color": "red",
        "price": 19.95
    },
    "expensive": 10
}

```

List of all books is required. You can define parameter 'books' with the value `input.List.store.book`.

This is an example expression to find if a parameter contains any expensive books:

```

for (var i=0; i<$.books.length; i++) {
    if ($.books[i].price>9) {
        'Expensive';
        break;
    }
    else {
        'Cheap';
    }
}

```

## 2 Branches

For each branch specify the expression value.

Default branch with undefined value is always available. It is used in case the expression results in not matching any other value defined on other branches.

## Output parameters tab

ION API Flow parameters created in the steps inside the decision flow cannot be referenced directly.

The **Output parameters** tab shows all API Flow parameters that are created in the steps inside the decision flow. It can be used as reference when specifying API Flow parameters as an input for the next steps. When two API Flow parameters with the same name are specified inside the decision flow. Both are displayed in the value column.

## Monitoring

Use ION API Gateway monitoring screens to track the runtime calls to an API flow. Each flow is registered as an API call, and a timeline indicates the individual APIs that were used.

---

**Note:** You must have the `IONAPI-Tracing` role to view the monitoring screen.

To monitor API flows:

- 1 Click the ION API icon in the Application Switcher panel.
- 2 Click the **Monitoring** tab on the left side.
- 3 You can filter events by API call specifications, including time and date. For more information, see the *Infor ION API Administration Guide*.

Each line item includes a drill-down link to view the flow of information that corresponds to the API flow.

Each API that is displayed in the flow also contains a drill-down for more details on the API call.

## Error handling

API flows are synchronous end to end. Therefore, the duration of the flow must be less than the global gateway time-out, which is five minutes.

The duration of each individual task must be less than one minute. If a step results in a failure, the API flow is stopped and an error message is displayed.

## Creating and using connection points

Connections points for integration with cloud applications are created by the system administrator depending on your entitlement.

You can use these connection points in data flows and you can edit their document configuration.

For the on-premises applications that require an integration through ION CE, you must create the connection points yourself. The on-premises applications communicate with ION CE through the Enterprise Connector.

**Note:** Connection points are not activated separately. A connection point is activated automatically when a data flow, that uses that connection point, is activated.

## Creating connection points from the data flow modeler

- 1 Select **Connect > Data Flows**.
- 2 Open a data flow or create a new one.
- 3 Add a connection point activity, such as **File**, to the flow.
- 4 In the properties pane, click **New**.
- 5 Select the type of connection point to be created. The **Connection Point Details** page is opened.
- 6 Specify the properties for the new connection point.

- 7 Click **Test** to test the configuration.
- 8 Save the connection point and click **Back** to return to the data flow details.

## Creating connection points from the connection points list

You can only create a connection point for on-premises applications if you have created a location first.

See [Creating a location](#) on page 65.

- 1 Select **Connect > Connection Points**. Existing connection points are displayed as tiles.
- 2 Click on the **Add** tile.

A list of Infor connection points and Technology connection points is displayed. The connection point types created through the Enterprise connector:

- **Infor Application**
- **Infor ERP LN/Baan**
- **File**

In addition to this, you have the **File (SFTP From Cloud)** type. This connection point is not created through the Enterprise connector but created as a cloud connection point which connects to your SFTP server.

- 3 Select the type of connection point to be created. The corresponding connection point details page is displayed.
- 4 Set the properties for the new connection point.

Each connection point must have a name. The name identifies the connection point. The connection point name can contain these characters:

- Characters ranging from a-z or A-Z
- Characters ranging from 0-9
- Underscore (\_)
- Dash (-)

Each connection point has a logical ID. Logical IDs are automatically generated based on the connection point type and name except for Cloud 2.0 connection points. For the Cloud 2.0 connections points the logical ID is retrieved from the Infor Cloud.

The Logical ID format is: `lid://infor.type.convertedName`

In the logical ID, the characters from the name are converted to lower case.

Points of attention:

- A Location field is displayed for connection points defined through an Enterprise connector. Specify the location where the application is deployed.
- Specify the connection parameters and click **Test** to verify connectivity. Using the **Test** button is only useful if the related Enterprise Connector is running.
- On the **Documents** tab, select the documents, their verbs, and where they are sent from and received in.

For the settings details of the **Infor Application** connection point type, see [Application connection points](#) on page 84.

---

For the settings details of the **File** connection point type, see the *Infor ION Technology Connectors Administration Guide*.

- 5 Save the connection point definition.
- 6 Navigate back to the overview page.

## Changing a connection point

You can reuse connection points in multiple data flows. Changing the properties of a connection point impacts all active data flows that use that connection point.

You can change the documents configuration of a connection point by adding or removing documents or verbs.

You cannot delete connection points or remove documents that are used in an active data flow.

To change a connection point:

- 1 Select **Connect > Connection Points**.  
Existing connection points are displayed as tiles.
- 2 Click the connection point tile. The details page opens.  
For connection points of cloud applications, you can only edit their document configuration.
- 3 Change the properties as required.
- 4 Save the connection point.
- 5 Navigate back to the overview page.




For the settings details of the **Infor Application** connection point type, see [Application connection points](#) on page 84.

For the settings details of the **File** connection point type, see the *Infor ION Technology Connectors Administration Guide*.

## Checking usage of connection points

- 1 Select **Connect > Connection Points**.  
Existing connection points are displayed as tiles.
- 2 Move the pointer over the connection point that you are interested in.
- 3 Click the [Usage](#) link in the bottom right corner of the tile.  
A list of data flows using the selected connection point is displayed. To open the details page of the selected object, double-click the row or select the row and click the details icon on the blue panel.  
Alternatively, open the connection point you require and click **USAGE** in the button bar on the details page.

This table shows the type icons:

Type	Name	Description	Status
	Data Lake Flow Name	Data Lake Flow Description	Data Lake Flow Status
	Document Flow Name	Document Flow Description	Document Flow Status
	Workflow Name	Workflow Description	Workflow Status

## Deleting a connection point

When a connection point representing a Cloud application is disabled by the system administrator, it has a **Pending Delete** status. It means that you can delete it from ION Desk. Enterprise Connector connection points and the **File (SFTP from Cloud)** connection point type can only be deleted if they are not used in a data flow.

To delete a connection point:

- 1 Select **Connect > Connection Points**.  
Existing connection points are displayed as tiles.
- 2 Click **Select** to start the multi-select mode. Select the tiles of the connection points to delete.
- 3 Click **Delete** to delete the selected connection points. Click **Cancel Select** to exit the multi-select mode.

To delete only one connection point, hover over its tile and click **Delete**.

**Note:** You cannot delete a connection point that is used in a document flow. Check the connection point usage and ensure that it is removed from all activities of document flows.

## File (SFTP from Cloud) connection point

The **File (SFTP from Cloud)** connection point defines a connection from ION CE to an external SFTP server.

The communication is direct and does not involve an Enterprise connector. You can create this connection point by providing the URL and credentials of the external SFTP file directory. Configurations in this connection point to send and receive documents are the same as the File connection point. For details about configuring a File connection point or a **File (SFTP from Cloud)** connection point, see the *Infor ION Technology Connectors Administration Guide*.

SFTP is a secure file transfer protocol of FTP over SSH. If you have an SFTP share open to the cloud, then ION CE acts as the SFTP client and can transport the files. Standard port 22 is used in this communication.

---

## Enterprise Connector

You can use the Enterprise Connector to integrate your local deployed applications with ION.

The Enterprise Connector is a local agent that is typically to be installed in your network.

Enterprise Connector provides an out-of-the-box connectivity with Infor Cloud through AWS SQS and S3 web services. It can transport files between ION in the multitenant Cloud, also known as ION CE, and your local applications.

**Note:** Enterprise Connector is not available in GovCloud because it is not certified for FedRAMP.

The Enterprise Connector service is deployed in your on-premises infrastructure and is responsible for the communication with the ION CE environment. The Enterprise Connector exchanges messages between ION CE and the on-premises applications that have a connection point defined related to this Enterprise Connector. Communication between the Enterprise Connector and ION CE is asynchronous.

The Enterprise Connector uses outbound connections to Amazon services that are exposed through https and port 443. Only an outbound connection is required. The Infor ION installation in the Infor cloud does not require any inbound connection to the Enterprise Connector.

Amazon SQS, S3 is used for intermediate storage.

A single Enterprise Connector can serve multiple on-premises applications. For performance reasons the Enterprise Connector service must be installed close to the applications for which it has a connection point running. We recommend that the Enterprise Connector service is installed in the same network segment (low latency).

You may have your applications that are distributed over multiple physical locations. Therefore, it is supported to have multiple Enterprise Connector services for the same ION CE tenant.

In ION CE an Enterprise Connector is related to a Location. To create connection points, that are related to on-premises applications, in ION CE, create a location first and install the related Enterprise Connector. Then create connection points that are related to that location.

## Integration with your local applications

You can use the Enterprise Connector to integrate your local deployed applications with ION. The Enterprise Connector currently supports these types:

- Infor Application connection point  
Defines a connection from ION to a local application that can send and receive messages. The messages are sent using Outbox tables and received using Inbox tables. ION connects to the Inbox and Outbox through JDBC.  
For configuration details, see [Application connection points](#) on page 84.
- IMS connection point  
Defines an ION Messaging Service based connection point that can send and receive messages. The communication is https and REST based.  
For configuration details, see [Application connection points](#) on page 84.
- Infor LN/Baan connection point

A special type of application connection point that connects to a local LN/Baan system. The connection is made from the Infor LN/Baan connection point created through the Enterprise Connector to the LN application. The typical connection information of Infor LN, such as the hostname, BSE environment, username, password, and Bshell, is used.

The **Maximum processing time per message** property in the Infor LN/Baan connection point defines the timeout for the processing of a single message. If there is no response from Infor LN/Baan after this time, processing of this message is stopped and a Confirm BOD is generated. ION continues with the processing of the next message. A timeout can happen if the message processing in LN is taking more time than the defined limit, or if the bshell does not respond anymore.

When using the channels to enable parallel processing, the maximum number of bshells allowed per channel is limited to 10.

For details about configuring an Infor LN/Baan connection point, see the *Infor LN Configuration Guide for Infor ION* in *InforXtreme Knowledge Base Article 22945150*.

- File connection point

Defines a connection from ION to a local application that cannot send or receive messages, but can create/process files. The connection is made from the File connection point through the Enterprise Connector to the local File application. It supports connecting to a local file server using FTP or SFTP and to a local Windows shared folder. For details about configuring a File connection point, see the *Infor ION Technology Connectors Administration Guide*.

- Database connection point

With the Database Connector you can connect applications that cannot send or receive documents, but have a database available.

In this way, you can retrieve data from a database and send it to one or more Infor applications. Or you can use data from Infor BODs to update your database.

There are two types of database connection point:

- Stored Procedure
- AnySQL Modeler

With the Stored Procedure type you can call stored procedures that are defined in your database, pass inputs, and receive output in the form of Infor BODs.

With the AnySQL Modeler you can create SQL select statements to read data from a database in visual modeler with minimum knowledge of SQL language. Learning specifics of each database protocol is not required. The same modeling experience is used for each of the supported database server types. Output is provided in the form of JSON documents.

- Message Queue Connection Point

You can use connectivity through message queues.

You can create a connection point to read or write a message queue. The Message Queue Connector of ION Enterprise Connector acts as a JMS client.

A connection point that is configured as a JMS Client, enables ION to directly connect to external JMS queues that are provided by other vendors. Messages are transported back and forth.

The use of this connection type requires a JMS BundleWrapper. For details, see the *Infor ION Technology Connectors Administration Guide*.



---

## Configuring an integration with a local application

- 1 Create a Location.
- 2 Download and install the Enterprise Connector service.
- 3 Configure and start the Infor ION Enterprise Connector service.
- 4 Create an Enterprise Connector connection point.  
Test the connectivity of the connection point with the local `InOutbox` database.
- 5 Incorporate the connection point in a data flow and activate the data flow.

## Creating a location

To create a location:

- 1 Select **Connect > Enterprise Locations**.  
A list of existing locations is displayed.
- 2 Click **Add**.  
The Location details window is displayed.
- 3 Specify Name and Description for the new Location and click **OK**. The **Enterprise Connector Location - Credentials** dialog box is displayed.
- 4 Click **show credentials**. Then, right-click and copy and paste the Location ID and Location Secret Key.  
Alternatively, click **download credentials** to store the credentials in a csv-formatted file.  
You will use these credentials later in the on-premises-deployed Enterprise Connector service to start communication with ION CE.
- 5 Click **Close** to return to the list. The location is now created. The **Connection Status**, **Version**, and **Host** fields are still blank.

## Downloading and installing the Enterprise Connector

The Enterprise Connector arranges the connectivity between ION in the cloud and your local application. The Enterprise Connector is running as a Grid service. Install it on a local Windows server, as close as possible to your application integration point. The exchange of messages is asynchronous through Amazon SQS and S3, ensure the Enterprise Connector service can access the Amazon services.

## Enterprise Connector prerequisites

System requirements for an Enterprise Connector server:

- Windows 2016, Windows 2019, or Linux Redhat Enterprise Server 7 and 8.  
Enterprise Connector is not supported on a Windows Domain Controller.

- Amazon Corretto JDK 8 (64 bits) update 212 or later installed. JDK 11 is not supported. It is, for example, known that the Bouncy Castle extension does not work well with Enterprise Connector. Therefore, you must use a default JDK installation without extensions.

Requirements for the database:

- SQL Server 2016
- SQL Server 2017
- PostGres Plus Advanced Server DB (PPAS), 64 bit version. Version 9.4, 9.5, 9.6, 10.6, 10, 11

Hardware specifications:

- 4 GB RAM, 2 Cores for low volume tests
- 8 GB RAM, 4 Cores for higher volume tests
- Enterprise Connector does not require much disk space because it does not store business data in its database. 10 GB of free space is sufficient.

ION CE Enterprise Connector 12 runs on the ION Grid. The ION Grid facilitates high availability for the Enterprise Connector and requires a database. During the installation you can either have the installer create the database or use a pre-created database. If the installer must create the database, a database user with database administrative authorizations is required. For SQL Server typically 'sa'.

During the installation of the Enterprise Connector, you must specify three ports. We recommend that you use 28089, 28090, and 28091. Ensure these ports are accessible for the host where the Enterprise Connector is installed. If you plan to scale-out the Enterprise Connector, ensure the ports are also accessible for the other hosts.

Ensure the Enterprise Connector can access these Amazon web services:

- IAM: endpoint <https://iam.amazonaws.com>
- SQS: the SQS endpoints for your region. For reference, see [http://docs.aws.amazon.com/general/latest/gr/rande.html#sqs\\_region](http://docs.aws.amazon.com/general/latest/gr/rande.html#sqs_region)
- S3: the S3 endpoints for all regions. For reference, see [http://docs.aws.amazon.com/general/latest/gr/rande.html#s3\\_region](http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region)

Ensure subdomains to these S3 endpoints are allowed; so, for us-east-1 allow \*.s3.amazonaws.com.

If you configure the Enterprise Connector with the AWS (Amazon) region it must access, you only must configure access to that region.

See [Limiting Enterprise Connector to the ION CE Amazon region](#) on page 75.

## Installing Enterprise Connector

For installations on Linux, see [Installing Enterprise Connector using a script](#) on page 68.

**Note:** You can install multiple Enterprise Connectors at the same host. To separate these installations, each installation requires its own database, its own port numbers, and a unique name for the installation folder. Ensure the server is sized to support the load of additional Enterprise Connectors.

### 1 Select **Connect > Enterprise Locations**.

A list of existing locations is displayed.

For the locations that have an Enterprise Connector service and the status is OK, the Version shows the version of the currently installed Enterprise Connector. If an update for that version is available, the version is colored red.

**2** Click the **Download Enterprise Connector** icon.

**3** Save the file: `com.infor.ion.cloud.enterprise.connector.installer-<version>.jar`

**4** Copy the `jar` file to the Windows system where the Enterprise Connector must be installed.

**5** Start the installation by double clicking the `jar` file.

Or run this command 'as Administrator':

```
Java -jar com.infor.ion.cloud.enterprise.connector.installer-<version>.jar
```

**6** During the installation you must specify this information:

- The Grid installation directory. This is defaulted to `c:\Program Files\Infor\IONEnterpriseConnector`
- Several TCP/IP ports for:
  - Bootstrap Service. We recommend 28089.
  - Host Router (https). We recommend 28090.
  - Host Router (proxy). We recommend 28091.
- The JDK path: Defaulted with the highest java version installed.
- The Database type: SQL Server or PostGres Enterprise DB.
- Grid Database Name: The default name is `InforIONEGrid`
- Database Scenario: Select one of these scenarios:
  - **Installer creates database.**  
The installer creates the Grid database. The ION Runtime Database User must exist. The installer gives this user full access to the Grid database.
  - **Install in pre-created empty database.**  
The empty Grid Database must exist. The ION Runtime Database User must exist and have full authorization on the Grid Database.
- The Connection Parameters to the database instance:
  - Database Server
  - Database Port or Database Instance
- ION Runtime Database User and Password.
- Database Administrator User and Password.  
Only required if you selected **Installer creates database**.
- Location Credentials and Tenant Mapping.  
Optionally, you can apply these properties after the installation:
  - Access Key ID
  - Secret Access Key
  - Tenant Mapping
- Proxy Service  
Optionally, you can apply these properties after the installation:
  - **Configure the use of a proxy service**

Select this check box if the Enterprise Connector must use a proxy service to access the Infor Cloud

- **Proxy Host (https)**  
Specify the host address for your proxy service.
- **Proxy Port (https)**  
Specify the port for your proxy service.
- **Authentication**  
Leave blank to use anonymous authentication.  
Select basic to use basic authentication.
- If basic authentication is selected, configure these properties:
  - **Userid**  
Specify the proxy userid.
  - **Password**  
Specify the proxy password and repeat it to verify the entry.

#### 7 Click **Next** to start the installation.

The installation process shows the status of the ION Enterprise Connector. The expected end status is OK.

**Note:** If you did not provide the Location credentials during the installation, the end status is NOT\_OK.

The log files for the Enterprise Connector are stored in C:\ProgramData\Infor\ION Enterprise Connector\install\installlog.

The Windows Services now include the **Infor ION Grid-<Installation folder name>** service.

## Installing Enterprise Connector using a script

For Linux: You must create a Linux user and group and link the user to the group. This user is used by Enterprise Connector to run the ION Grid bootstrap daemon. Example:

```
groupadd ionec
adduser ionec -g ionec
```

- 1 Verify if the required system prerequisites are met.
- 2 We recommend that you install in /opt. Prepare an `installer.properties` file with these parameters:

```
# Infor ION Scriptable Install Input template.
# -----
#
# Grid Installation Configuration
grid.installation.dir=/opt/infor/IONEC
# Only for Linux
```

```

grid.service.username=ionec
grid.service.group=ionec

# Grid Connection Configuration
bootstrap.service.port=28089
host.router.https.port=28090
host.router.port=28091

# JRE Path (JDK is required)
jrePath=/usr/java/jdk1.8.0_65

# Database Selection
# =====
# database.type = 'sqlserver' or 'postgres_ppas'
# database.scenario = 'nodb' or 'precreateddb'
# nodb means the installed will create the database # requires
# database.admin.username and database.admin.password to be set #
# precreateddb means the database (grid.database.name) already exists,
# as an empty database. database.type=sqlserver grid.database.name=InforI
# ONECGrid database.scenario=precreateddb

# SQL Server database properties
database.server= MySQLSERVERDATABASE
database.instance=
database.port=1433
#database.admin.username=sa
#database.admin.password=****
ionruntime.database.username=ecruntime
ionruntime.database.password=ecruntimepwd

# Location Credentials and Tenant Mapping - Optional
# =====
iam.access.key.id= AKIBJP3YJYRMFA3AOHXA
iam.secret.access.key= WsfDtVAXnGX2vd2+svE9qC8oItALLD2tvD2e5otA
tenant.mapping=INFOR

```

### 3 Run the installer using the just created `installer.properties` file as input.

```

java -jar com.infor.ion.cloud.enterprise.connector.installer-version.jar
installer.properties

```

## Downloading the latest Enterprise Connector version programmatically

To download the latest version of the Enterprise Connector, you can use rest services that are exposed by the current Enterprise Connector, version 12.0.42 or later. For this download a configured Enterprise Connector is required. The download uses the location credentials to pick up the latest version from the Infor cloud environment. To call these methods, a grid-authenticated identity with the `grid_pro`

visioner security role is required. This can for example be a client certificate or an OAuth 10.a key-pair.

- GET `https://<echost:port>/api/provision/v1//enterpriseconnector/version`  
This method provides the `installedVersion` and `CloudVersion`. You can compare them to see whether a later version is available.
- GET `https://<echost:port>/api/provision/v1/enterpriseconnector/download`  
Use this method to start the download of the cloud version.
- GET `https://<echost:port>/api/provision/v1/enterpriseconnector/download/status`  
Use this method to get the status of the download and the location of the downloaded installer.

#### Example script to create an identity

- PKCS12 client certificate with `grid_provisioner` role, password **ecpassword**, valid for 90 days:  

```
java -jar <ECInstallFolder>\tools\grid-cli.jar cert createClient -ClientKsDir=c:\temp -ClientName=ecdownload -ClientPassword=ecpassword -keystoreType=PKCS12 -roles=grid-provisioner
```
- Oauth 10.a key-pair:  

```
java -jar <ECInstallFolder>\tools\grid-cli.jar oauth10 create -description="downloadEC" -roles=grid-provisioner -name=ECdownload
```

For more information on creating a client certificate or Oauth key-pair, see the *Infor ION Grid Security Administration Guide*.

## Configuring Enterprise Connector

To configure the Enterprise Connector service:

- 1 Go to the Windows system where the Enterprise Connector service is installed.
- 2 Open the Grid Management UI.
  - a Run the file `<Installation folder>\bin\AdminUI.cmd`
  - b Accept the certificate warning.
  - c You can run `AdminUI.cmd` with another Windows user than the one used to install ION Enterprise Connector. In that case you must add your user to the local Windows Group `IONEnterpriseConnector_full` or `<installfolder>_full`  
To add the user on Windows, start `lusrmgr.msc`
- 3 In the Grid Management, click **Applications > IONEnterpriseConnector > Management Pages**.
- 4 Specify the location credentials you retrieved from the ION CE Location screen by either importing the credentials csv file or pasting the ID and Security Key; click **Load**.  
The location information for this location is retrieved from ION CE and displayed.

- 5 If your on-premises applications exchange messages using another tenant than the ION CE tenant, define the on-premises tenant in the **Tenant Mapping** field. The Enterprise Connector converts the tenant of the messages from cloud to on-premises tenant and vice versa.  
**Note:** You can only change the tenant mappings if there are no active connection points. To change the mapping when there are active connection points, you must deactivate the related document flows to deactivate these connection points. Then, change the tenant mapping and activate the document flows again.
- 6 Click **Save**.  
The Enterprise Connector service is updated to use the configuration.

## HTTP security settings

- 1 Open the Grid Management UI.
- 2 Select **Security > Security**. The **Security** page contains an **HTTP Security Setting** section. With ION Enterprise Connector 12.0.40 and later, all these settings are enabled by default.
- 3 Pay specific attention to this setting:

### Host Header Validation

You can use a load balancer or other indirections in front of the Enterprise Connector. To do so, you must add the host address (FQDN) that is used at the client side to the white list for this setting. Otherwise requests that are posted by the client are rejected.

For more information, see the online help of this security page.

## Granting other users access to the Enterprise Connector configuration

If Enterprise Connector is installed on a Windows platform, then by default only the user that installed Enterprise Connector has access to the installation folder.

To grant access to other users:

- 1 Run `lusrmgr.msc`.
- 2 Select **Local Users and Groups > Groups**.
- 3 Look for the group name of the installation folder of Enterprise Connector. The default name is `IONEnterpriseConnector_full`.
- 4 Open the group and add the users as members.

## Accessing the Enterprise Connector Grid Management UI remotely

When you deploy the Enterprise Connector on a Linux host, this step is required. For a Windows host deployment, this step is optional because you can also open the Grid Management UI locally.

To access the Grid Management UI from a remote browser:

**1 Create a client certificate for authentication:**

- a For Windows, create a copy of this sample script:

```
rem Create Grid client certificate for provisioning
set INSTALLDIR=C:\Program Files\Infor\ ION Enterprise Connector
set CLI=%INSTALLDIR%\tools\grid-cli.jar
set CLIENTDIR=C:\Temp
java -jar "%CLI%" -dir "%INSTALLDIR%" cert createClient -clientname
  "Grid Management UI access" -keystoreType "PKCS12" -roles "grid-
  admin" -clientKsDir "%CLIENTDIR%" -clientPassword ChangeMe
```

For Linux, create a copy of this sample script:

```
#create Grid client certificate for provisioning
INSTALLDIR=/var/infor/IONEnterpriseConnector
CLI=${INSTALLDIR}/tools/grid-cli.jar
CLIENTDIR=/tmp
java -jar ${CLI} -dir ${INSTALLDIR} cert createClient -clientname
  "Grid Management UI access" -keystoreType "PKCS12" -roles "grid-
  admin" -clientKsDir ${CLIENTDIR} -clientPassword ChangeMe
```

- b Make these changes to the script:

- Set `INSTALLDIR` to the installation folder of Enterprise Connector.
- Set `CLIENTDIR` to the folder where you want to create the keystore.
- Change the client password, *ChangeMe*, to your preference.

- c Run the script.

**2 Prepare your browser.**

- a Import the client certificate keystore, generated in the previous step, in your browser keystore. The procedure to do this depends on the browser.
- b In your browser, open this URL:
- ```
https://<hostname>:<host router https port>/grid/ui/soho
```
- c A warning, that the SSL certificate of the Grid is not trusted, is displayed. Accept the exception.

## Trusting the Enterprise Connector SSL certificate

The Enterprise Connector Grid exposes a self-signed certificate. If you must connect remotely through HTTPS to the Enterprise Connector, this certificate must be trusted by the clients. This applies , for example, in these situations:

- You want to access the Enterprise Connector management remotely.
- You have an application that integrates through ION Messaging Service (IMS).

You can export the public key and have that key trusted at the client. Alternatively, you can use a certificate that is signed by a trusted certificate authority (CA).



To export the public key of the self-signed certificate, open the Enterprise Connector Management UI. Then export the public key of the certificate through your browser.

To configure a trusted CA certificate in the Enterprise Connector:

- 1 Go to the Windows system where the Enterprise Connector service is installed.
- 2 Open the Grid Management UI.
  - a Run the `<Installation folder>\bin\AdminUI.cmd` file.
  - b Accept the certificate warning.
  - c You can run `AdminUI.cmd` with another Windows user than the one used to install ION Enterprise Connector.

To do so, you must add your user to one of these local Windows groups:

- `IONEnterpriseConnector_full`
- `<installfolder>_full`

To add the user on Windows, start `lusrmgr.msc`.

- 3 In the Grid Management Pages, select **Security > Identities** and select the **Show Default Identities** check box.

For details on this procedure, see "Managing HTTPS identities in the Grid Management Pages" in the *Infor ION Grid Security Administration Guide*.

## Enterprise Connector Configuration Screen

### Import Location Credentials button

Imports the location credential `.csv` file, downloaded from the ION Desk Location credentials dialog box.

### Take Location button

This button is only available if the credentials of the location are already taken by another Enterprise Connector installation. Use this button to transfer the ownership of a location from one Enterprise Connector Service to another. You typically use this in recovery scenarios.

Ensure that the Enterprise Connector you take the credentials from is stopped.

### Load button

Contacts ION CE, based on the specified Location credentials, to retrieve location information for this Enterprise Connector. Used when the initial credentials are provided. And used to update the configuration information if the location description in ION CE is changed.

### Location Access Key ID and Security Key field

Alternative for the import to specify the Location ID and Security key.

### Tenant Mapping

If your on-premises applications exchange messages using another tenant than the ION CE tenant, define the on premises tenant. In that situation, the Enterprise Connector converts the tenant of the messages from cloud to on-premises tenant and vice versa. If the field is empty, which is the default, the Cloud tenant is set and expected in the message headers by the Enterprise Connector.

---

## Location information

### Name

Location Name as configured in ION Desk Location screen.

### Description

Location description as configured in ION Desk Location screen.

### Status:

- OK: Connectivity with ION CE is OK
- ERROR: Connectivity with ION CE fails. A reason for failure is displayed. Possible reasons for failure:
  - Not able to connect to Amazon. Verify whether your server has access to the Internet, specifically if it can access Amazon services.
  - Incorrect credentials: The Location credentials are invalid. Apply correct credentials and, if required, recreate the credentials through the ION Desk location dialog box.
  - Location locked by <hostname>: The Location credentials are already used by another Enterprise Connector. To overrule this, you can use the **Take Location** option, but be careful because that takes away the ownership from the other Enterprise Connector.

### Connection Points

The number of connection points that are configured for this Enterprise Connector.

### Tenant

The ION CE tenant id. This tenant identifier must be used by the on-premises applications when they exchange messages through the Enterprise Connector. Unless an 'on-premises tenant' is defined.

### JMS Client Drivers button

Opens the JMS Client Drivers screen.

### Custom Database Drivers button

Opens the Custom Database Drivers screen

## JMS Client Drivers Screen

On this screen the JMS drivers are managed that are used by this enterprise connector. Open the screen through the main screen of Enterprise Connector Management.

### Custom JMS Client libraries

You can use the Message queue connection point as a client to connect to a JMS vendor. In that case you must upload the related JMS client library. Click **Download JMS Driver Wrapper** to download the bundle wrapper. This bundle wrapper program generates the deployment bundle of your dynamic JDBC driver/ JMSclient library. After a bundle is generated click **Import JMS Client Driver** to upload it to ION Service. You can also remove an existing bundle when it is not used by any connection point. Note that when you remove an existing bundle, the Enterprise Connector Service is restarted. This results in a temporary outage of the Enterprise Connector until the service is started again.

Instructions on using the JMS driver wrapper, see *Infor ION Technology Connectors Administration Guide*

## Custom Database Drivers Screen

On this screen the Database drivers are managed that are used by this enterprise connector. Open the screen through the main screen of Enterprise Connector Management.

### Custom Database Drivers

If you are using the Infor application connection point with a database vendor that is not supported. You can upload the JDBC driver of your vendor into ION. Use the bundle wrapper program. You can retrieve the program by clicking **Download Custom Database Driver Wrapper**. This bundle wrapper program generates the deployment bundle of your dynamic JDBC driver. After you have created the bundle wrapper, you can upload it. Click **Import Custom Database Driver**.

**Note:** When you add or remove a bundle, the Enterprise Connector Service is restarted. This results in a temporary outage of the Enterprise Connector until the service is started again.

See *Infor ION Technology Connectors Administration Guide*

## Limiting Enterprise Connector to the ION CE Amazon region

When the Enterprise Connector starts to connect to Amazon, it first must find out the region it must access. Therefore, the Enterprise Connector requires access to all Amazon regions. To limit the access to only the Amazon region that is used by the Enterprise Connector, you can configure the `aws.region` property.

- 1 Open the Enterprise Connector Grid Management UI.
- 2 Select **Applications**.
- 3 In the **IONEnterpriseConnector** tile, open the menu by clicking the hamburger icon. Then select **properties**.
- 4 In the search box, specify `aws.region` and click **Search**.  
Click the <undefined> value and specify as value the Amazon region of the ION CE you use.  
Possible values:
  - us-east-1
  - eu-central-1
  - eu-west-1
- 5 Restart the Enterprise Connector application.
  - a Open **Applications**.
  - b At the EnterpriseConnector node, click **Stop**.  
The node restarts automatically.
- 6 If it was configured already with location credentials, verify connectivity is still working.

## Managing the Grid by REST services

Various REST services are available to monitor the Enterprise Connector status and Grid status.

---

See `https:// <enterpriseconnectorhost>:<host.router.https.port>/grid/rest/api-docs/ui/index.html`

An example of a useful method:

```
GET https:// <enterpriseconnectorhost>:<host.router.https.port>/grid/rest/applications
```

In the result for the ION Enterprise Connector, to see if the application state is fine, verify if the value for the `globalState:state = OK`.

## Uploading a JMS driver

You can upload a JMS driver through a REST based webservice. The wadl for this service is available at:

```
https://<enterpriseconnectorhost>:<host.router.https.port>/api/provision/application.wadl
```

## Updating the JDK version

Complete one of these steps:

- The procedure to switch can happen as part of the upgrade to the latest Enterprise Connector version:
  - a Download and install the Amazon Corretto JDK 8 update 212 or later on the ION Enterprise Connector host.
  - b Download the latest ION Enterprise Connector installer and run the installer.
  - c In the installation dialog boxes, at the point where you must select the JDK, specify the location of the Corretto JDK.
  - d Continue the installation.
- Alternatively, to update the JDK version of an ION Enterprise Connector host, see "Changing the JDK for a grid host" in the *Infor ION Grid Administration Guide* .  
If you have deployed ION Enterprise Connector over multiple hosts, complete this procedure for each host.

## Updating location credentials

You can create new location credentials and apply these in the Enterprise Connector.

- 1 Select **Connect > Enterprise Locations**.
- 2 Select the Location for which new credentials are required and click **Recreate Credentials**. The **Enterprise Connector Location - Credentials** dialog box opens.

- 3 In this dialog box, either click **show credentials** and copy and paste the Location ID and Location Secret Key or **download credentials** in a csv formatted file.
- 4 Click **Close** to return to the list. The status of the Location is ERROR because the current Enterprise Connector has invalidated credentials.
- 5 Open the **Grid Management UI**, on a server where the Enterprise Connector is installed.
- 6 Select **Applications > IONEnterpriseConnector > Management Pages**.
- 7 Specify the location credentials you retrieved from the ION CE Location screen by either importing the credentials csv file or pasting the ID and Security Key. Click **Load**. The location information for this location is retrieved from ION CE and displayed.
- 8 Save the changes and verify whether the status is OK.

## Log archiving

The Enterprise Connector logs are archived to the database automatically every Saturday and Sunday. Logs in the archive that are older than 60 days are deleted.

To download archived logs or adjust the archive schema:

- 1 Open the Grid Management UI.  
See [Configuring Enterprise Connector](#) on page 70.
- 2 Select **Monitoring > Log archiver**.  
On this page, you can download archived logs or adjust the archiving configuration.

## Configuring the use of a proxy service

If you use a proxy service to route Internet communication, you must configure the proxy service settings in ION Enterprise Connector. After this is set, only https communication between Enterprise Connector and the Infor Cloud, that is, Amazon web services, is routed through the proxy service. All other https communication, such as the IMS communication, does not use the proxy configuration.

- 1 Open the Grid Management UI.  
See [Configuring Enterprise Connector](#) on page 70.
- 2 Navigate to **Applications** and select the **IONEnterpriseConnector** application.
- 3 Click the hamburger icon and select **Properties**.
- 4 In the **enterprise.connector.aws.proxy** property section, specify the properties as appropriate for your proxy service.

### **aws.proxyHost**

Specify the host address for your proxy service.

### **aws.proxyPort**

Specify the port for your proxy service.

**aws.proxyAuthenticationMethod**

Leave blank if you use anonymous authentication.  
Specify `basic` if you use basic authentication.

If basic authentication is selected, configure these properties:

**aws.proxyUsername**

Specify the proxy username.

**aws.proxyPassword**

Specify the proxy password.

## Scale-out the Enterprise Connector to another host

For availability reasons you can add another host to the Enterprise Connector deployment. In case one of the Enterprise Connector hosts fails, the Enterprise Connector continues to function by running at the remaining host. Ensure the hosts you use can communicate with each other and have TCP/IP access. Enable access to these ports:

- `hostrouter.https.port`
- `bootstrap.port`
- `hostrouter.port`

To scale-out the enterprise connector to another host:

- 1 Another host must be available that complies to the prerequisites for an Enterprise Connector server.

You are supposed to use the same OS type for each Enterprise Connector, so either Linux or Windows.

Ensure to install the same JDK version in the same folder as on the original host.

If you use a different version, you must adjust the `jdk.path` in the properties file.

For linux deployment: ensure you create user and group with the same name on the original host. Otherwise you must adjust `service.group` and `service.username` in the properties file.

- 2 Copy these files to the new host:

- `installer-version.jar`
- `grid.install.properties`

The files are located in these folders:

- **Windows:** `c:\ProgramData\Infor\ION Enterprise Connector\<InstallFolder>\Grid Installer`
- **Linux:** `/var/log/infor/ion enterprise connector/<InstallFolder>/GridInstaller`

- 3 Specify this information in the `grid.install.properties` file:

**grid.hostname**

The hostname of the new host

**grid.hostaddress**

The full qualified domain name of the new host.

**database.password**

The password of the database user defined in `database.username`.

Example properties file for Windows:

```
#Thu Oct 27 17:19:09 CEST 2016
grid.hostname=IONCEHOST
jdk.path=C:\Program Files\Java\jdk1.8.0_91
grid.hostaddress=IONCEHOST.acme.com
install.path=C:\Program Files\Infor\IONEC
database.instanceName=
database.name=InforIONECGrid
database.password=*****
database.type=sqlserver
database.username=ecruntime
service.userType=virtual
grid.properties.0=grid.internal.featureFlags\=appendBcByDefault\=false

database.port=1433
hostrouter.https.port=28090
bootstrap.port=28089
database.host=DATABASEHOST
database.schema=
hostrouter.port=28091
sqlserver.datasource.multiSubnetFailover=false
```

- 4 Go to the new host and run the grid installer with this command:

```
java -jar installer-version.jar grid.install.properties
```

## Uninstalling an Enterprise Connector host

To uninstall the Enterprise Connector from a host, run this command:

```
java -jar <installation folder>/uninstall/uninstall.jar
```

After running this command at the last host, you have uninstalled the Enterprise Connector.

## Enterprise Connector service - processing

The Enterprise Connector service starts polling and pushing messages for a specific application when the related connection point to Enterprise Connector in ION is activated.

Data flow activation and deactivation have related start and stop polling commands. These commands are transmitted from ION in the cloud to the local Enterprise Connector through Amazon SQS and S3. This is an asynchronous process. An actual activation or deactivation is picked up and executed by the Enterprise Connector when it is running and connected to Amazon SQS and S3.

When one or more IMS connection points are activated for this Enterprise Connector, an extra process is started at the Enterprise Connector host. This is, the ION Messaging Service node. For the ION Messaging Service node one instance is running. If the Enterprise Connector is scaled to multiple hosts, the Enterprise Connector Grid decides where that instance runs. The Enterprise Connector is restarted on the other (failover) host if the current host goes down.

You can find logging and configuration information for the Enterprise Connector service at each host in these folders:

- `<Installation Folder>\grid\log\SYSTEM`  
This folder contains logfiles for the core grid functionality.
- `<Installation Folder>\grid\log\IONEnterpriseConnector`  
This folder contains logfiles for the core enterprise connector node.
- `<Installation Folder>\grid\log\IONMessagingService`  
This folder contains logfiles about the processing of incoming IMS requests to the Enterprise Connector.

## Verifying Enterprise Connector

You can run several checks in the ION Desk and the **Grid Management UI** to verify the Enterprise Connector.

In ION Desk you can check the status of:

- Enterprise Connector Location  
Open **Connect > Enterprise Location**. Verify whether the **Connection Status** is **OK** and the **Version** is not displayed in red. Indication the Enterprise Connector is at the latest version. When you move the pointer over the value of the **Hosts** field, the tooltip shows more details about the hosts where the Enterprise Connector is deployed.
- Enterprise Connector connection points  
**Connect > Active Connection Points**. Verify the Status is **OK**.
- A particular message sent to a connection point of the Enterprise Connector .  
Open **OneView**. Search for the message. In the message timeline, check whether the enterprise connection point shows the *message delivered by any adapter or engine* event.

In Grid Management UI, at the Enterprise Connector side, you can open the:

- **Hosts** tab and verify each Grid host has the *Started* status.
- **Applications** tab and verify if the `IONEnterpriseConnector` has the **OK** status.  
On the **ION Enterprise Connector** application tile, click **Management Pages**. Verify if the **OK** status is shown. Check if the number of the active and paused connections points meet your expectation.



---

Information about managing the ION Grid and running an application in the ION Grid, see these guides:

- *Infor ION Grid Administration Guide*
- *Infor ION Grid Security Administration Guide*

## Verifying Enterprise Connector connectivity

Through ION Desk you can check if ION can communicate with your local deployment.

### 1 Select **Connect > Enterprise Locations**.

For each location the Status is displayed.

### 2 Check this information:

#### **Status**

Every 30 seconds the Enterprise Connector sends a heartbeat to ION CE to determine the status of the connectivity.

- empty: This location is not yet allocated by an Enterprise Connector.
- ERROR: The location is allocated by an Enterprise Connector, but there is an issue with the connectivity to this Enterprise Connector. Verify whether the Enterprise Connector Service is running and the configuration is correct, that is, valid Location credentials are used. To verify this, start the Configuration screen of the Enterprise Connector.
- OK: The location is allocated by an Enterprise Connector and connectivity with that Enterprise Connector service is OK.

#### **Host**

Identifies the server(s) where the Enterprise connector is running.

#### **Version**

Identifies the version of the installed Enterprise Connector. The version is colored red if the version is outdated. In that case, download the Enterprise Connector version from the location screen and install it on the Enterprise Connector server.

### 3 List the connection points using a particular location. Select the location and click **Usage**.

## Upgrading Enterprise Connector from an earlier 12 version

The Infor Cloud team can update ION CE to a later version. After such an update, the Enterprise Connector version in the ION Desk **Enterprise Location** page is red. This indicates a later version is available for you. For details on what this newer version offers, see the ION CE release notes.

Follow this procedure to update your installed version 12 Enterprise Connector to a later version. During the upgrade, the integration between ION CE and the connection points for this location is down. You must run the Enterprise Connector upgrade at one of the Enterprise Connector hosts.

### 1 In ION Desk, select **Connect > Enterprise Locations**.

- 2 Click the **Download Enterprise Connector** icon and save the `com.infor.ion.cloud.enterprise.connector.installer-<version>.jar` file.
- 3 Copy the jar file to the system where the Enterprise Connector must be upgraded. If you have scaled out the Enterprise Connector, select one of the hosts.
- 4 Start the installation by double-clicking the jar file.  
Alternatively, run this command:  

```
Java -jar com.infor.ion.cloud.enterprise.connector.installer-<version>.jar
```

  
For Windows, you must run this command 'as Administrator'.
- 5 During the upgrade, you must specify this information:
  - The Grid installation directory.  
Default: `C:\Program Files\Infor\IONEnterprise Connector`  
Select the installation folder of the current Enterprise Connector.
  - The JDK path.  
Specify the JDK used currently.  
Alternatively, to update the used JDK version, specify a later version.
  - ION Runtime Database Password.
- 6 After the upgrade has finished, verify whether the Enterprise Connector service is running and message processing continues.

## Upgrading using a script

### Note:

ION Enterprise Connector version 12.0.38 and later write their version in this file:

- **Windows:** `C:\ProgramData\Infor\ION Enterprise Connector\<InstallFolder>\info.json`
- **Linux:** `/var/log/infor/ion enterprise connector/<InstallFolder>/info.json`

- 1 Prepare an `upgradeinstaller.properties` file with these parameters:

```
# Infor ION Scriptable Upgrade Install Input template.
# -----
-
# Grid Installation Configuration
# Makes sure grid.installation.dir points to the current Enterprise
Connector directory
grid.installation.dir=/opt/infor/IONEnterpriseConnector
# Make sure ionruntime.database.password contains the correct password
related to the ion enterprise connector runtime database user
ionruntime.database.password=ecruntime
```

- 2 Run the installer using the `upgradeinstaller.properties` file as input.

```
java -jar com.infor.ion.cloud.enterprise.connector.installer-version.jar upgradeinstaller.properties
```

---

## Disaster recovery planning

If your Enterprise Connector installation is corrupted, you can perform these actions:

- Replace Enterprise Connector with a fresh installation.
- Recover the Enterprise Connector database.

### Replacing Enterprise Connector with a fresh installation

The Enterprise Connector only stores configuration data in its database. No transactional data is stored in the database. Therefore, if your Enterprise Connector installation, including its database, is corrupted, you can revive it by completing these steps:

- 1 Install a fresh ION Enterprise Connector deployment.
- 2 Reapply the location credentials for the original location.
- 3 Configure the correct tenant mapping.

During this process a message is displayed to inform you that the location is already in use. Use the **Take Location** option to transfer the ownership for the location to this Enterprise Connector.

### Recovering the Enterprise Connector database

Apply this procedure to recover the Enterprise Connector including its configuration.

Ensure you have a backup of these components:

- ION Enterprise Connector Grid database
- These files in `c:\programdata\Infor\ION Enterprise Connector\GridInstaller`:
  - `grid.install.properties`
  - `installer-<version>.jar`

**Note:** Create a new backup if you have upgraded the Enterprise Connector.

To perform the recovery, complete these steps at a fresh host with the same name as the original host:

- 1 Recover the database.
- 2 Recover the `grid.install.properties` and `installer-<version>.jar` files.
- 3 In the `grid.install.properties` file, make these adjustments:
  - `database.password= #` - Specify the correct password for the database user that is defined in the `database.username` property.
  - `grid.hostname= #` - Change the name so it deviates from the original entry. The name does not have to be a DNS entry.
  - Ensure the `bootstrap.service.port` has another value than before.
  - Ensure the `host.router.port` has another value than before.

- 4 Run this command:

```
java -jar installer<version>.jar grid.install.properties
```

- 5 After the installation, open the Grid management UI and verify the Enterprise Connector.

## Application connection points

An application connection point defines a connection between ION and an application that can send and receive messages in the format defined in ION Data Catalog. ION Data Catalog is also known as ION Registry.

Application connection points are of type IOBOX or ION Messaging Service (IMS).

IMS integration can be a direct integration or through ION API.

This table shows the characteristics of these connection point types:

| Type                        | Description                                                                                                                                                                                                      |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IOBOX                       | Sends messages using outbox tables and receives messages using in-box tables. The in-box and outbox are connected through JDBC. Only messages of type BOD are supported with this connection point.              |
| ION Messaging Service (IMS) | Exchanges messages with an application through predefined REST methods. This type of connection point authenticates through OAuth 1.0. IMS supports all document types that are defined in the ION Data Catalog. |
| IMS via ION API             | Exchanges messages with an application through predefined IMS REST methods. Connectivity and authentication is handled by ION API.                                                                               |

The Infor applications typically offer this way of connecting to ION. Other applications can also adopt this.

For details, see the *Infor ION Development Guide*.

## Configuring an IMS integration through ION API

Before you can configure an IMS integration between an application and ION, through ION API, you must prepare the application for this communication.

For details, see the *Infor ION API Administration Guide*.

To learn the IMS interface interaction, see "Infor Application Connector (IMS)" in the *Infor ION Development Guide*.

## Bidirectional IMS through ION API

Bidirectional IMS through ION API, application has IMS end point:

- 1 In ION API, create an Authorized App of the Backend Service type.  
This authorized app represents the client side of the application, that is calling ION through ION API.
  - a During the creation ION API generates a Client ID for this application. You must specify this ID in the connection point you create in step 3c.
  - b As part of the creation, you must download credentials, during this process.  
You are prompted to specify a user name. This is not required because ION does not authorize incoming requests based on user authorizations.  
You require the downloaded credential file in step 4a, where you configure the application.
- 2 In ION API, create through Available APIs a new API Suite, with these points of attention.
  - a Specify the **Suite Name**. You must select this name later in the IMS through ION API connection point; see step 3a.
  - b Ensure the APIContext name is unique. Do not change this later because you reference it in your application.
  - c At the Endpoint details, specify this information:
 

**Target Endpoint URL**  
Specify the URL including the IMS path for your service.

**Proxy Endpoint URL**  
For example public. This becomes part of the URL path that is called by ION IMS.

**Proxy Security**  
Select **OAuth 2.0**.  
ION IMS uses OAuth 2.0 authentication to access your service.

**Target Endpoint Security**  
Select the authentication method your application expects.
  - d At the Endpoint Documentation, specify the documentation for your IMS APIs. The preferred option for ION API is through swagger documentation.
  - e Verify whether, based on the documentation, the IMS methods are listed.
- 3 In ION Desk, create the IMS through ION API connection point, with these points of attention
  - a For the IMS Endpoint, select the product suite you created in step 2.
  - b For the Service Account, import a service account you created through User Management / Service Accounts.
  - c For the ION API Client Id, specify the client ID that was generated in step 1a.
- 4 At the Application side, use the Client ID credentials that were created in step 1a.

For details, see the *Infor ION API Administration Guide*.

## Unidirectional IMS through ION API

Unidirectional IMS through ION API, application has no IMS end point:

- 1 In ION API, create an Authorized App of the Backend Service type.

This authorized app represents the client side of the application, that is calling ION through ION API.

- a During the creation ION API generates a Client ID for this application. You must specify this ID in the connection point you create in step 3.
  - b As part of the creation, you must download credentials, during this process.  
You are prompted to specify a user name. This is not required because ION does not authorize incoming requests based on user authorizations.  
You require the downloaded credential file in step 3, where you configure the application.
- 2 In ION Desk, create the IMS through ION API connection point, with these points of attention
    - a Clear the **Application has IMS End Point** check box.
    - b For the ION API Client Id, specify the client ID that was generated in step 1a.
  - 3 At the Application side, use the Client ID credentials that were created in step 1.

For details, see the *Infor ION API Administration Guide*.

## General application properties

In addition to the name, logical ID (which are relevant for any connection point type), an application connection point has this property:

| Type            | Description                                                                                                                                                                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Logical ID Type | The type to be used to construct the middle part of the logical ID. For example, if the logical ID type is 'wm' and the name is 'Warehouse1' then the logical ID is: 'lid://infor.wm.warehouse1'. The logical ID type must contain characters ranging from a-z or 0-9 or an underscore (_) or a dash (-). Capitals and spaces are not allowed. |

## Infor application (IMS) properties

This table shows the available buttons:

| Button                       | Description                                                                                                                      |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| TEST                         | Click this button to test connectivity with the IMS application endpoint.                                                        |
| IMPORT AND TRUST CERTIFICATE | Click this button to import the https SSL certificate of the IMS application endpoint if this certificate is not trusted by ION. |

This table shows the available properties:

| Property                             | Description                                                                                                                                                                                                                                                                                          |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application has IMS End Point        | If this check box is selected, the IMS connection point can send and receive messages.<br>If this check box is cleared, <b>IONTEST</b> button, <b>IMPORT AND TRUST CERTIFICATE</b> can only receive messages from the application. The button, and <b>IMS Endpoint</b> properties are not available. |
| IMS Endpoint – URL                   | IMS can only receive URL for the application to integrate with                                                                                                                                                                                                                                       |
| IMS Endpoint – OAuth Consumer Key    | OAuth consumer key for the application to integrate with                                                                                                                                                                                                                                             |
| IMS Endpoint – OAuth Consumer Secret | OAuth secret key for the application to integrate with                                                                                                                                                                                                                                               |
| ION Endpoint – Endpoint Path .       | Display field. Shows the path to be used by the integrated application to run the ION IMS methods. The <code>host:port</code> must be replaced with the actual <code>host:port</code> of the ION Enterprise Connector host and Host Router (https) port or the load balancer pointing to it.         |
| ION Endpoint – OAuth Consumer Key    | Display field. Shows the OAuth consumer key that must be used by the integrating application                                                                                                                                                                                                         |
| ION Endpoint – OAuth Principal       | Display field. Shows the OAuth Principal name as registered in the Infor OS Grid.                                                                                                                                                                                                                    |

The application you configure to integrate with ION through IMS must trust the SSL certificate of the Enterprise Connector installation.

See [Trusting the Enterprise Connector SSL certificate](#) on page 72.

## IMS via ION API properties

This table shows the available text buttons:

| Button                              | Description                                                                                                                      |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| <b>TEST</b>                         | Click this button to test connectivity with the IMS application endpoint.                                                        |
| <b>IMPORT AND TRUST CERTIFICATE</b> | Click this button to import the https SSL certificate of the IMS application endpoint if this certificate is not trusted by ION. |

This table shows the available properties:

| Property                           | Description                                                                                                                                                                                                                                                                                                      |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Application has IMS End Point      | If this check box is selected, the IMS via ION API connection point can send and receive messages.<br>If this check box is cleared, ION can only receive messages from the application. The <b>TEST</b> button and <b>IMPORT AND TRUST CERTIFICATE</b> button and the IMS Endpoint properties are not available. |
| IMS Endpoint - Product & Operation | Display field. Shows the selected IMS discovery endpoint. Select the IMS discovery endpoint from the application that must integrate with ION through IMS.                                                                                                                                                       |
| IMS Endpoint - Service Account     | The service account that is used by ION to authenticate for the application IMS endpoint.                                                                                                                                                                                                                        |
| Authorization - ION API Client Id  | The ION API Client Id of the application as registered in ION API. Used by the ION connection point to verify whether the received message is sent by the application. The Client Id represents the application.                                                                                                 |

## Infor application properties

This section is only applicable for the connection points of type Enterprise Connector.

On the Connection tab of the application connection point you can select the Infor Application type. This table shows the available properties:

| Property        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database Driver | Select one of the supported database types. See the supported database table below                                                                                                                                                                                                                                                                                                                                                                   |
| Database        | Select one of these configurations for the database address: <ul style="list-style-type: none"> <li>Basic - specify properties for the database. The properties to be specified depend on the selected Database Type. The URL to connect to the database will be built automatically.</li> <li>Advanced - specify the URL to connect to the database. Use this if you must use advanced settings that are not shown when selecting Basic.</li> </ul> |
| Hostname        | Specify the host name of the server that hosts the database. This field is only enabled if the <b>Basic</b> checkbox is selected.                                                                                                                                                                                                                                                                                                                    |
| Port Number     | Specify the number of the port that must be used to connect to the database. This field is only enabled if the <b>Basic</b> checkbox is selected.                                                                                                                                                                                                                                                                                                    |
| Schema Name     | Specify the name of the database schema to be used. This field is only enabled if the <b>Basic</b> checkbox is selected and the Database Type is SQL Server, Oracle DB2 or DB2/400.                                                                                                                                                                                                                                                                  |
| SID             | Specify the name of the Oracle schema ID to be used. This field is only enabled if the <b>Basic</b> checkbox is selected and the Database Type is Oracle                                                                                                                                                                                                                                                                                             |



| Property                  | Description                                                                                                                                                                                                                                                              |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| URL                       | Specify the URL of the database that contains the Inbox and Outbox tables. The URL is dependent of the database type. See the supported database table below for details.<br><br>If you selected the Basic configuration for the database address, the URL is read-only. |
| User Name                 | Specify the user name to connect to the database. Note: when using SQL Server with Windows authentication, specify a valid Windows domain username instead of a SQL Server user name.                                                                                    |
| Password:                 | Specify the password to connect to the database.                                                                                                                                                                                                                         |
| Delete Processed Messages | Whether processed messages must be deleted immediately from the Outbox after processing them.                                                                                                                                                                            |

This table shows the supported database types and the URL formats:

| Supported Database Drivers | URL format                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Oracle                     | <code>jdbc:oracle:thin:@//[hostname]:[port]/[oracle_sid]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| DB2                        | <code>jdbc:db2:// [hostname]:[port]/[schemaname]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| DB2/400                    | <code>jdbc:as400://[hostname];naming=system;libraries=[schemaname];prompt=false</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| MS_SQLServer               | <p>The URL is specified in this format: <code>jdbc:sqlserver://[hostname[instanceName]][:port]][;property=value[;property=value]]</code></p> <p>For example, when using SQL Server authentication on a default instance with the <b>MultiSubnetFailover</b> property of a cluster setup: <code>jdbc:sqlserver://[hostname]:1433;databaseName=[schemaName];MultiSubnetFailover=true</code></p> <p>For a complete connection URL specification of this driver, see the Microsoft documentation at <a href="https://msdn.microsoft.com/en-us/library/ms378428(v=sql.110).aspx">https://msdn.microsoft.com/en-us/library/ms378428(v=sql.110).aspx</a>.</p> |
| Postgres (PPAS)            | <p><code>jdbc:postgresql://[HostName]:[PortNumber]/[SchemaName]?stringtype=unspecified</code></p> <p>For a complete connection URL specification of this driver, see the Postgres documentation at: <a href="https://jdbc.postgresql.org/documentation/93/connect.html">https://jdbc.postgresql.org/documentation/93/connect.html</a></p> <p>For using this driver in ION, the “?stringtype=unspecified” parameter is required.</p>                                                                                                                                                                                                                    |

When you create an Infor application connection point with DB2 as the database type, the user account used in the connection properties must equal the user account which created the Inbox/Outbox tables in the database.

If you want to reuse this I/O box for multiple Logical IDs, see [Configuring Connection points for a Multi-Logical ID I/O box](#) on page 90.

## Configuring Connection points for a Multi-Logical ID I/O box

Applications such as Infor CloudSuite Industrial have individual connection points per site configured in ION and use the same In-box and Outbox table between them. In such situations, ION must process messages based on Tenant and based on Logical ID. To enable this, you must create an additional column in the I/O box tables. In addition to this, a property called `By LogicalID` is added to the category of **Message Processing preference in I/O box** in the **Advanced Properties** section of the **Application connection point**.

To share the same I/O box across multiple Logical IDs:

- 1 Run the I/O box script `<db>_upgrade.sql` from the 3.0 folder in your I/O box.  
For more information, see "Download Scripts" in "Configuration Files tab".
- 2 Ensure that a new column called `C_LOGICAL_ID` is added to both `COR_OUTBOX_ENTRY` and `COR_INBOX_ENTRY`
- 3 In ION Desk, go to **Connect > Connection Points**.
- 4 Double-click an application connection point.
- 5 On the **Connection** tab expand the **Advanced** section.
- 6 Find the category called **Message Processing preferences in I/O box**.
- 7 Set **By Tenant** and **By LogicalID** to **True**.  
You must select the Tenant before you can set the value for **By Logical ID**.
- 8 To publish a message from your application, specify this information:
  - In the `COR_OUTBOX_ENTRY` table, specify a value for `C_LOGICAL_ID`
  - In the `COR_OUTBOX_HEADERS` table, specify the `FromLogicalID` key with the correct Logical ID value. This ID must match the Logical ID value specified in the connection point.
- 9 Use the connection point in your data flow and activate it.

**Note:** ION matches the Tenant and the Logical ID value in a case-sensitive manner. To avoid inconsistencies in tenant processing, you must specify the Tenant ID and the Logical ID according to Infor standards. For the Infor Tenant ID standards, see "Configuring Connection points for a Multi-Tenant I/O box" in the *Infor ION Desk User Guide*.

The Infor Logical ID standards are:

- ID must always start with the prefix "lid://".
- ID must contain characters ranging from a-z or 0-9 or an underscore (\_) or a dash (-). Capitals and spaces are not allowed.

## Application documents

In the **Documents** tab of an application connection point, select the documents that can be sent or received by the application. The selected documents must match the actual capabilities of the application.

---

Some application connection points only support documents to be `Sent from Application` or `Documents to Write`.

To add a document type, click **Add**. A list of documents that are based on available `verb.noun` combinations is displayed. You can filter the list of documents by specifying a text in the **Filter** field. Alternatively, use the switch buttons to show the Infor standard documents or custom documents.

In the **Documents** tab you cannot select the `Acknowledge` and `Show` response verbs. The verbs are not visible in this tab but are automatically supported if a relevant request document is selected in the other direction. For example, when `Process.SalesOrder` is selected as `Send from Application`, the `Acknowledge.SalesOrder` is automatically considered as supported in `Receive in Application`. The data flow with this connection point can consume `Acknowledge.SalesOrder`.

**Note:** For a unidirectional IMS connection point ("Application has IMS End Point" is disabled) it is allowed to configure the `Acknowledge` and `Process` response BOD types.

To remove a document or verb, select the item and click **Remove**. When removing a document or verb that is used in a data flow, an error is reported in the data flow notifications.

Some application connection points can discover the documents automatically. The add and remove documents options are not available for these connection points. Click **Refresh** to start the auto discovery. During the discovery, if the application exposes documents not known to the ION registry, these documents are ignored. A warning, that shows which documents are ignored, is displayed.

## Network connection point

A Network connection point defines a connection between two tenants in one multi-tenant environment. A Network connection point can send and receive any messages defined in Data Catalog.

## Configuring a Network connector

- 1 Go to the **Documents** tab of a Network Connection point.
- 2 Select the documents that can be sent to or received from other tenant.
- 3 To add a document type click **Add**.

A list of documents is displayed. You can filter the list of documents in these ways:

- Specify a text in the **Filter** field.
- Select a document type.
- Use the switch buttons to show the Infor standard documents or custom documents.

- 4 To remove a document, select the item and click **Remove**.

When removing a document that is used in a data flow, an error is reported in the data flow notifications.

## Tenant – Tenant dependency

The selected documents must match the selection in the other tenant. If the document is selected as *Send to Tenant* the same document must be selected in the other tenant as *Receive from Tenant* and the other way around. The document is not accepted by the target side and a *ConfirmBOD* is created on the source side in these cases:

- A document is selected as ‘Send to Tenant’ on the Source side and it is not selected in the ‘Receive from Tenant’ list of active network connection point on the Target side.
- A network connection point is not active on the target side.

## Document types supported by various connectors

This table shows the document types supported by ION for each connector:

| Connection Point Type          | BOD | JSON | ANY | DSV |
|--------------------------------|-----|------|-----|-----|
| Application using Inbox/Outbox | ✓   | ✗    | ✗   | ✗   |
| Application using IMS          | ✓   | ✓    | ✓   | ✓   |
| Message Queue                  | ✓   | ✗    | ✗   | ✗   |
| Network                        | ✓   | ✓    | ✓   | ✓   |
| Database                       | ✓   | ✗    | ✗   | ✗   |
| File                           | ✓   | ✓    | ✓   | ✓   |
| Web service                    | ✓   | ✗    | ✗   | ✗   |
| Oracle BS Connector            | ✓   | ✗    | ✗   | ✗   |
| SAP Connector                  | ✓   | ✗    | ✗   | ✗   |
| ION API                        | ✓   | ✓    | ✓   | ✗   |

## Document mappings

Use a mapping in the data flow to transform the contents` of a document or to transform a document into another type of document. How to create a mapping see [ION Connect Mapping](#) on page 115.

---

When using a mapping in a data flow, note that:

- A mapping cannot combine two BODs into one or split one BOD into multiple BODs. The latter can be achieved with a parallel flow having a specific mapping in each branch.
- Using mappings for reply BODs (Acknowledge or Show) is not supported. Reply BODs cannot be modeled in a data flow.
- Be careful when using mappings for request BODs (Process or Get). Reply BODs are directly sent from the receiver of a Process BOD back to the original sender of the Process BOD. For example, if you change the document type in a mapping. The sender of the Process BOD receives an Acknowledge for another document type than expected. If you change the verb from Process to another verb, the sender does not receive an Acknowledge. If you change the verb Sync to Process, the sender of the Sync BOD receives an unexpected Acknowledge message. Do not change the document type or the verb in case of a request BOD.

To use a mapping in a data flow:

- 1 Open a data flow or create a new one.
- 2 Add a mapping activity to the flow
- 3 Select the mapping to be used.

The resulting document after the mapping activity is filled automatically based on the To Document and To Verb of the selected mapping.

## Splitter

You can use a splitter in the data flow to split one document that contains multiple instances of the same object into separate documents. Each document with one object together with a document header and footer.

For example, when a Purchase Order with multiple Purchase Order Lines is received and as the next step an API must be called per line. Splitter can be used to split a Purchase Order into multiple Purchase Orders. Each one containing all headers and footers, and only one Purchase Order Line. API is then called once per each created Purchase Order.

Supported document types for splitting are BOD and JSON Conventional. To define a required splitting element, you must use Xpath or JSON Path, respectively.

## Using a splitter in a data flow

- 1 Click **Connect > Data Flows**.  
A list of existing data flows is displayed.
- 2 Open a data flow or create a new one.
- 3 Add a splitter activity to the flow.
- 4 Specify a name and description.
- 5 Click the **Configuration** tab.

- 6 For each listed document, specify the path expression to identify where the document must be split.
  - a Click the magnifier glass.
  - b Select element in the **Select Attribute** window.
  - c Click **OK**.
 Or specify the required path manually.

## Rules for using a splitter in a data flow

When using a splitter in a data flow, these rules apply:

- One output document is created for each instance of element defined by the path.
- The message ID of the output documents is the same as the Message ID of the input document but extended with a sequence number. The sequence number is for example:1 or :2
- If a defined path is not found in an input document, no splitting is done. The input document is passed to the output as is. The Message ID is extended with :0
- If an input document is split into at least two output documents, the instances message header is deleted.
- All other message headers are copied from the input document to all output documents.
- A sibling to a parent element without any instance of element that is defined by a path, is not included in any output document. See Example 2 - Excluding parent element.

### Example 1 - Basic splitting logic

- 1 Input Purchase Order with MessageID = 123456789 and this simplified structure:

```
<SyncPurchaseOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <PurchaseOrder>
      <PurchaseOrderHeader> ... </PurchaseOrderHeader>
      <PurchaseOrderLine> ... </PurchaseOrderLine>(1)
      <PurchaseOrderLine> ... </PurchaseOrderLine>(2)
      <PurchaseOrderLine> ... </PurchaseOrderLine>(3)
    </PurchaseOrder>
  </DataArea>
</SyncPurchaseOrder>
```

- 2 Configured to split at this line:

```
SyncPurchaseOrder/DataArea/PurchaseOrder/PurchaseOrderLine
```

- 3 These three instances of <PurchaseOrderLine> element are found:

- Document with MessageID = 123456789:1 and this simplified structure:

```
<SyncPurchaseOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <PurchaseOrder>
      <PurchaseOrderHeader> ... </PurchaseOrderHeader>
      <PurchaseOrderLine> ... </PurchaseOrderLine>(1)
    </PurchaseOrder>
  </DataArea>
</SyncPurchaseOrder>
```

- Document with MessageID = 123456789:2 and this simplified structure:

```
<SyncPurchaseOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <PurchaseOrder>
      <PurchaseOrderHeader> ... </PurchaseOrderHeader>
      <PurchaseOrderLine> ... </PurchaseOrderLine>(2)
    </PurchaseOrder>
  </DataArea>
</SyncPurchaseOrder>
```

- Document with MessageID = 123456789:3 and this simplified structure:

```
<SyncPurchaseOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <PurchaseOrder>
      <PurchaseOrderHeader> ... </PurchaseOrderHeader>
      <PurchaseOrderLine> ... </PurchaseOrderLine>(3)
    </PurchaseOrder>
  </DataArea>
</SyncPurchaseOrder>
```

## Example 2 - Excluding parent element

- 1 Input document with multiple parent elements:

```
<SyncSalesOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <SalesOrder> (A)
      <SalesOrderHeader> ... </SalesOrderHeader>
```

```

<SalesOrderLine> ... </SalesOrderLine>(1)
<SalesOrderLine> ... </SalesOrderLine>(2)
</SalesOrder>
<SalesOrder> (B)
  <SalesOrderHeader> ... </SalesOrderHeader>
</SalesOrder>
</DataArea>
</SyncSalesOrder>

```

**2** Configured to split at this line:

SyncSalesOrder/DataArea/SalesOrder/SalesOrderLine

**3** These two instances of SalesOrderLine element are found:

- Document with MessageID = 123456789:1 and this simplified structure:

```

<SyncSalesOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <SalesOrder> (A)
      <SalesOrderHeader> ... </SalesOrderHeader>
      <SalesOrderLine> ... </SalesOrderLine>(1)
    </SalesOrder>
  </DataArea>
</SyncSalesOrder>

```

- Document with MessageID = 123456789:2 and this simplified structure:

```

<SyncSalesOrder>
  <ApplicationArea> ... </ApplicationArea>
  <DataArea>
    <Sync> ... </Sync>
    <SalesOrder> (A)
      <SalesOrderHeader> ... </SalesOrderHeader>
      <SalesOrderLine> ... </SalesOrderLine>(2)
    </SalesOrder>
  </DataArea>
</SyncSalesOrder>

```

**4** This information is NOT forwarded to the next steps:

```

<SalesOrder> (B)
  <SalesOrderHeader> ... </SalesOrderHeader>
</SalesOrder>

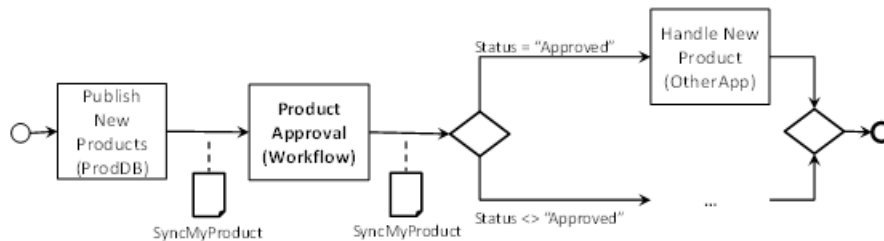
```



## Workflow from document flow

You can use a workflow activity to start a workflow as part of a document flow. For example, you can retrieve data on new products from an application database. Then run a workflow to review and approve the new product. After the new product is approved send it to another application for subsequent processing.

The results of the workflow can be incorporated in the document.



In this example, the initial document can be:

```
<SyncMyProduct>
  <ApplicationArea>
    ...
  </ApplicationArea>
  <DataArea>
    <Sync>
      ...
    </Sync>
    <MyProduct>
      <ID>1234</ID>
      <Description>New Product</Description>
      <Link>http://somewhere/products?1234</Link>
    </MyProduct>
  </DataArea>
</SyncMyProduct>
```

The product ID, Description and Link can be used as input for the workflow. The new Status (either Approved or Rejected) can be output of the workflow. After the workflow is completed, this document is sent to the next step in the flow:

```
<SyncMyProduct>
  <ApplicationArea>
    ...
  </ApplicationArea>
  <DataArea>
    <Sync>
      ...
    </Sync>
    <MyProduct>
```

```

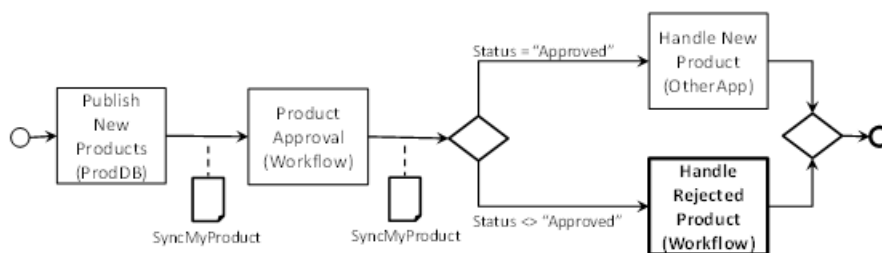
<ID>1234</ID>
<Description>New Product</Description>
<Link>http://somewhere/products?1234</Link>
<Status>Approved</Status>
</MyProduct>
</DataArea>
</SyncMyProduct>

```

In the workflow activity you select the workflow definition and link the input and output parameters of that workflow to attributes of the incoming document. You can pass structured information from the document (for example order lines) as input data for the workflow structures. The workflow structures cannot be used as output.

When a document arrives at this activity, a workflow is started. When the workflow is completed, the output of the workflow is added to the document and the document flow continues.

A workflow activity can also be used at the end of a document flow. In that case the workflow output (if any) is ignored. For example:



Multiple workflow activities can be used in a document flow. A workflow can be used in multiple workflow activities in a single document flow or in multiple document flows.

## Modeling a workflow from document flow

- 1 Select **Connect > Data Flows**.
- 2 Open the details page of the document flow for which you must add the workflow activity.
- 3 Click **Workflow** to add the activity to the flow from the toolbox.
- 4 Ensure the workflow activity has a preceding activity and the preceding activity provides one document type. The verb of the document must be either Sync or Process. The type of output document of the workflow activity is the same as the input document
- 5 In the workflow activity properties, select the **Properties** tab.
- 6 Specify the name for the activity and optionally add a description.
- 7 Select the workflow to be used. Click **New** to create a new workflow definition or click **Details** to view the details of an existing workflow definition.

The workflow parameters and workflow structures from the workflow definition are retrieved when the workflow is selected or the **Refresh Workflow** button is used.

**8** Select the **Attributes** tab.

**9** Select any attributes of the document that must be used in the workflow activity.

See [Selecting attributes](#) on page 103.

You can only select attributes after a document is selected as on the line between the preceding activity and the workflow activity.

**10** Select the **Parameter Mapping** tab.

Here you can select which attributes to use for each input or output parameter of the workflow.

You can only select an attribute if the data type matches the data type of the workflow parameter. The name of the attribute can differ from the name of the workflow parameter. For each input parameter, an attribute must be selected. If the workflow activity is followed by a next activity in the flow you must select an attribute for each output parameter.

**11** Select the **Structure Mapping** tab. Observe that the root of the workflow structure is mapped to the noun name from the document that triggers the workflow activity. Mapping the other elements of the structure is optional. To complete the mapping, click **Map** next to each structure element.

You can map levels from the workflow structures to document attributes that have children. Mapping is done top down. Sub-levels or fields from the structure can only be mapped to child elements that are based on the parent level mapping. The mapping process is done through a selection dialog box and allows the selection of valid elements that have matching data-type.

Optionally, you can specify a filter for the selected attributes. Click **Filter** next to the path you mapped. The xml-attributes of the selected element and the xml-attributes of its repeating parent elements are displayed. Edit the filter by specifying a value for one of the attributes and click **OK**. The path is updated with the filter. At runtime, the occurrences that comply with the filter are extracted from the document and mapped into the workflow structure. If there are several occurrences of the document element mapped to one single field, the first occurrence is selected.

**Note:**

- a** Skipping several levels when you map a field to a document element is called a "mapping with elevation". If several sibling fields are mapped using a mapping with elevation. Use a filter to ensure the fields are selected from the same path determined by the filter.
- b** Fields in workflow structures are single occurrence. To retrieve all occurrences when the document element mapped to the field has multiple occurrences. You must define a workflow structure level that is mapped to the direct parent of the document element. In other words, avoid mapping with elevation in this case.
- c** You cannot map a flat structure from a BOD, for example a custom BOD, into a more complex workflow structure, with several levels.

**12** When using a workflow activity, the document flow depends on the workflow.

This means:

- When changing the input or output parameters of a workflow, you must update the workflow activity accordingly. Use the **Refresh** button from the **Properties** tab to get the latest set of input and output parameters and updated structures of the workflow. Note that if a structure is changed, all mappings are removed upon Refresh.
- Activate the workflow definition before activating the document flow. You cannot deactivate the workflow definition as long as the document flow is active.

- When the running workflow is canceled or fails, the workflow activity raises a Confirm BOD on the incoming document. Resubmitting this from the Error BODs page starts a new instance of the workflow.
- If one or more workflows are running for a workflow activity, it is advisable to not deactivate the document flow. Otherwise the results of the workflow cannot be handled when the workflow completes. In this case an error is logged, because the result of the workflow cannot be delivered to the workflow activity anymore.
- When exporting and importing the document flow, also export and import the used workflow definition.

## Workflow from document flow process

The `ProcessWorkflow` BOD is used to start a workflow from a workflow activity. You do not have to model or send this message. In ION the `ProcessWorkflow` BOD is automatically created based on the modeled workflow activity.

When a document is received by a workflow activity, these actions are taken:

- 1 The values of the attributes that are required as input for the workflow, are extracted from the document.  
**Note:**
  - If the selected attribute occurs multiple times in the document then only the first value is used as input for the workflow.
  - If the selected attribute does not occur in the document then the workflow is started without a value for the input parameter. This only works if the workflow input parameter has 'Use Initial Value if Null' selected in the workflow modeler.
  - If repeating attributes are mapped to a structure, the workflow structure contains as many levels as repetitions. This depends on the mapping definition and if filtering is modeled.
- 2 The request to start the workflow (`ProcessWorkflow`) is built and sent to the workflow engine.
- 3 The workflow is started and a copy of the original document is saved.
- 4 After the workflow completes, the output of the workflow, the `AcknowledgeWorkflow`, is sent back to the workflow activity.
- 5 If no next activity exists in the document flow then the output of the workflow is ignored.

If a next activity exists then the copy of the original document is retrieved. The output parameters from the workflow are merged into the document and the document is sent to the next step in the document flow.

**Note:**

- If the attribute that is linked to the workflow output parameter already exists then the existing value is overwritten.
- If the attribute does not yet exist then it is added to the document.
- If the attribute occurs multiple times in the document then only the first value is updated.
- Structures cannot be used as output, they do not change the document.

---

## Filtering and content-based routing

A connection point can send documents. The document flow defines which types of documents, such as sales orders or purchase orders, must be delivered to which other connection points. Even for a single document type, not all document instances are relevant for the next activity in a flow. Based on the content of the document, documents may or may not be needed. Use filtering or content-based routing to avoid delivering too many documents to a connection point.

In the toolbox of the document flow modeler, two types of flows are available for this purpose:

- **Filter**

Depending on the document content and the filter conditions specified, messages are sent to a destination if the filter condition is fulfilled, or ignored.

For example, the `CodeDefinition` BOD represents multiple types of code master data. An application can only be interested in a subset of all code definitions, for example only 'reason codes' are relevant.

The filtering is not limited to connections having only one document. Multi-document flows, such as master data flows or reporting flows, can also be filtered. You do not have to create multiple flows in that case.

- **Routing**

Depending on the document content, messages of the same type are distributed to one or more destinations, depending on the routing conditions. If none of the routing conditions is met, documents are ignored.

For example, a warehouse management application represents only one warehouse per application instance. An ERP application sends `SyncPurchaseOrder` messages. Depending on the warehouse on the purchase order line, the message must be sent to one or more specific warehouse management instances. A message is only sent to application instance `WMS_1` if it contains at least one line for warehouse.

**Note:** To define filters or routing, the metadata for the documents to be routed must be available in the Registry.

If the filter or routing receives BODs with the same document name but different verbs, the filtering or routing is the same for these verbs.

**Note:** The content-based routing depends on the data that is available in the BOD. If the sending application does not set attributes that are used in the condition, the result is not as expected. The contents of the BOD can also depend on the used verb and action code. For example, if a Sync message is sent with `actionCode Delete`, the document can only contain the identifiers, such as the document ID. To deal with such situation, use the `actionCode` in a condition or include a condition that explicitly checks for the existence of an attribute.

Content-based routing and filtering do not change the contents of a BOD. Either the complete BOD is passed on or not. Note that:

- A document in the BOD can include repeating elements, such as multiple sales order lines. If the condition includes a check on such lines, the BOD is passed on if at least one of the lines matches the condition.

- For some verbs, a BOD can contain multiple document instances, such as multiple sales orders. In that case, the BOD, including all document instances, is passed on if at least one of the sales orders matches the condition.

## Filtering

To add a filter:

- 1 Click **Connect > Data Flows**.  
A list of existing data flows is displayed.
- 2 Select the document flow where the filter must be added
- 3 Click **Details** to open the document flow modeler
- 4 Add a **Filter** activity to the flow.
- 5 If the preceding activity provides more than one document type, use **select for documents** as filter.
- 6 Click the **Attributes** tab;
- 7 Select any attributes of the document that must be used in the filter.  
See [Selecting attributes](#) on page 103
- 8 Click the **Conditions** tab.
- 9 Specify the conditions for the selected attributes.  
See [Defining conditions](#) on page 105. After activating the flow, the ION Service ensures that messages are filtered according to the specified conditions.

### Notes:

- Multiple flows can be defined using the same connection points and document types. In ION Service those flows using a logical disjunction ('or') are combined. For example; an active flow orders that `SyncSalesOrder` documents having status X must be sent from A to B. Another active flow orders that `SyncSalesOrder` documents having status Y must be sent from A to B. Then all `SyncSalesOrder` documents having status X or Y are sent from A to B.
- Filtering is done at message level. Either the whole message is delivered or nothing. The message is not partly delivered (for example by only including the relevant purchase order lines).
- When, using attributes that occur multiple times in a document the document is passed on if at least one of the lines matches the filter. For example, attributes from an order line can occur multiple times.
- The `actionCode` in BODs is not changed by the filtering. If the first BOD with `actionCode Add` is filtered out, and the second BOD for the same object instance with `actionCode Update` is passed. The `actionCode` is `Update`.
- Filtering is not applied to BODs used Event Management. If an application publishes `SyncSalesOrder` and an event monitor exists for `SalesOrder`. Then all `SyncSalesOrder` BODs are delivered to the event monitor, even if in the document flow a filter is specified.

## Routing

To use content-based routing:

- 1 Click **Connect > Data Flows**.  
A list of existing data flows is displayed.
- 2 Select the document flow to add the content based routing to.
- 3 Click **Details** to open the document flow modeler.
- 4 Add a **Routing** activity to the flow.  
Note that a routing can only be used if the preceding activity produces one type of document.
- 5 Right-click the diamond shape of the routing activity to add or remove branches. By default, a routing activity has two branches.
- 6 Click the **Attributes** tab.
- 7 Select any attributes of the document that must be used in the filter.  
See [Selecting attributes](#) on page 103.
- 8 Specify the conditions on the selected attributes.  
See [Defining conditions](#) on page 105.
- 9 Select which condition must be used for each branch.  
After activating the flow, the ION Service ensures that messages are filtered according to the specified conditions.

## Selecting attributes

In a filter or routing step you can select the document attributes that must be used in the conditions. You can select document-specific attributes such as an order status or planned delivery date. Additionally you can select one or more predefined attributes if needed.

These are the available predefined attributes:

- **AccountingEntity**: the ID of the accounting entity to which the document belongs.
- **Location**: the ID of the location to which the document belongs.
- **ActionCode**: a code to indicate the sender's view of what happened to the document(s) that are included in the BOD. For request BODs such as Sync or Process the `actionCode` is *Add*, *Change*, *Replace* or *Delete*. For `Acknowledge` BODs the action code is *Accepted*, *Modified* or *Rejected*. Show BODs do not contain an action code.

To add an attribute select at least one attribute of the monitor's application documents. You can use the selected attributes in the monitor rule, the monitor conditions, or the alert message.






## To select attributes

- 1 Click **Connect > Data flows**.

A list of existing document flows is displayed.

- 2 Click **Details**.
- 3 Click **Add** to display the **Select Attributes** window.
- 4 Expand the tree and select the desired attributes.

The selected attributes are displayed in the tooltip of the “Selected” count at the top of the window. This table shows the attribute types that can be selected in the **Select Attributes** window.

Attribute type	Icon	Description
Data Element		An element in a business document that can have a value. For example, an Amount element. A data element can be selected. Note that if the data element is contained in a repeating group, the data element can occur multiple times in a business document. A data element can contain attributes. To display the attributes, expand the data element node.
Repeating Element		A special type of data element that can occur multiple times in a business document. For example, a Note element. A repeating element can be selected. A repeating element can contain attributes. To display the attributes, expand the repeating element node.
Attribute		An attribute in a business document that can have a value. For example, a currencyID that is linked to an Amount element. An attribute can be selected. <b>Note:</b> If the data element is contained in a repeating element or in a repeating group, the attribute can occur multiple times in a business document.
Group		A grouping element in a business document that does not have a value, but contains other group elements or attributes. For example, a SalesOrderHeader. A group has children, such as data elements, attributes, or other groups. To display the children, expand the group node.
Repeating Group		A special type of group that can occur multiple times in a business document. For example, a SalesOrderLine. A group has children, such as data elements, attributes, or other groups. To display the children, expand the repeating group node.

**Note:** There are three predefined attributes that can be selected from the document header information: `AccountingEntityID`, `LocationID` and `actionCode`. Use these attributes if the delivery of a document to a next step in the flow depends on the document characteristics. For example: to only deliver documents from a specific accounting entity, include a comparison condition based on the attribute `AccountingEntityID` in the content-based routing or filter.

- 5 Click **OK**.
- The selected attributes are displayed in the **Attributes** tab.



- 6 You can change the names of the attributes in the **Attributes** tab.  
For example, specify the name of the application document, to which the attributes belong, as a prefix. This is useful if you selected attributes from multiple documents.

## Specifying filters for selected attributes

This procedure is optional.

Filter column in the list of attributes is not related directly to the filter condition. It is used to indicate that the selected attribute as a specific filter.

Data Elements and Attributes that are part of a repetitive structure can require additional filtering to be used in a condition. For example, the Location from the "ShipToParty" information on a Contract application document can occur several times. To specify which Location address details must be used in the condition evaluation. You can add a filter on the attribute "type" of the Location element.

To specify a filter:

- 1 Select **Connect > Data Flows**.
- 2 Click the **Attributes** tab.
- 3 Add one or more attributes that belong to a repeating group. StreetName.  
For example: `Contract/ContractHeader/ShipToParty/Location/Address/`
- 4 Select the attribute.  
This enables the **Filter** button.
- 5 Click **Filter** to display the **Edit Filter** page.  
The xml-attributes of the selected element and the xml-attributes of its repeating parent elements are displayed.
- 6 Specify the attribute to use for filtering. For example, **type**.
- 7 Click the **Equals Value** column and specify a value. For example **Office**.
- 8 Click **OK**.  
The path of the monitor attribute is updated with the filter condition. For example: `Contract/ContractHeader/ShipToParty/Location[@type="Office"]/Address/StreetName`

## Defining conditions

A condition defines which documents are relevant and which ones are not.

- In a filter step, the used condition is selected. If that condition is met (in other words, it evaluates to 'true') then the document is passed on to the next activity. If the condition is not met (it evaluates to 'false') then the document is not passed on.
- In a routing step, a condition is selected for each branch. Consequently a document is sent to a branch if the defined condition is met (in other words, it evaluates to 'true').

To specify a condition for a filter or for the routing branches:

**1** Select **Connect > Data Flows**.

A list of existing document flows is displayed.

**2** Click the **Conditions** tab.**3** Click **Add** to add a new condition.**4** Specify a name for the condition.**5** Select the condition type.

These types of conditions can be used:

- Attribute-value comparison, to check an attribute value. For example, **Status = Open**.
- Attribute comparison, to compare two attribute values. For example, **Discount > StandardDiscount**.
- Attribute Existence, to check whether a given attribute exists.
- Combined conditions, to combine two other conditions using **or** or **and**. For example, **Status = Open and Discount > StandardDiscount**.

These condition types and the properties to be set are described in more detail later.

**6** Click **OK** to close the dialog.**7** Add more conditions, if combined conditions are needed.

For example, first create an attribute-value comparison condition, then add an attribute comparison condition and then add a combined condition to combine these two.

**8** Select the used condition for the filter or for each of the routing branches.

## Attribute-value comparison

Use this comparison to compare the value of an Application Document attribute to a fixed value. This type of comparison is useful for attributes whose value does not change frequently.

This table shows the syntax:

<b>Syntax</b>	[Attribute][Operator][Value]
<b>Attribute</b>	One of the Application Document attributes that are selected for monitoring. The attribute can be of type: <ul style="list-style-type: none"> <li>• String</li> <li>• Integer</li> <li>• Decimal</li> <li>• Date and Time</li> <li>• Date</li> <li>• Time</li> <li>• Boolean</li> </ul>
<b>Operator</b>	Comparison operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Value</b>	A constant value of the same type as the attribute.

<b>Evaluation</b>	The evaluation result is True or False.
-------------------	-----------------------------------------

### Example

You build a condition to monitor contracts that have started after January 1st 2005. In the condition you use the `StartTime` attribute from the Contract Application Document.

This table shows the syntax:

<b>Attribute</b>	<code>StartTime</code>
<b>Operator</b>	<code>&gt;</code>
<b>Value</b>	<code>2005-01-01T00:00:00</code>

The evaluation result of the condition is true if the Date and Time value of the `StartTime` is after January 1st 2005 00:00 hours.

## Attribute comparison

Use this comparison to compare values of two Application Document attributes that are selected for monitoring. You can compare attributes from the same Application Document or from different Application Documents. The attributes must have the same data type.

This table shows the syntax:

<b>Syntax</b>	<code>[Attribute1] [Operator] [Attribute2]</code>
<b>Attribute 1</b>	One of the Application Document attributes that are selected for monitoring. The attribute can be of any type.
<b>Operator</b>	Comparison operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Attribute 2</b>	One of the Application Document Attributes that are selected for monitoring. The attribute is of the same type as Attribute 1.
<b>Evaluation</b>	The evaluation result is True or False.

### Example

Build a condition to verify if a partial shipment has occurred for a sales order line. In the condition compare the `OrderedQuantity` attribute from the `SalesOrder` Application Document with the `ShippedQuantity` attribute from the `Shipment` Application Document.

<b>Syntax</b>	<code>[Attribute1] [Operator] [Attribute2]</code>
<b>Attribute 1</b>	<code>OrderedQuantity</code>
<b>Operator</b>	<code>&lt;</code>

<b>Attribute 2</b>	ShippedQuantity
--------------------	-----------------

The condition evaluates to true when the comparison between the values of the two attributes is true.

## Attribute existence

Use this condition type to check whether an attribute has a value in an incoming document.

This table shows the syntax of the attribute existence condition type:

<b>Syntax</b>	[ Attribute] [Check]
<b>Attribute</b>	One of the document attributes that are selected for monitoring. The attribute can be of any type.
<b>Check</b>	One of these values: <ul style="list-style-type: none"> <li>Exists The evaluation result is True if the attribute has a value.</li> <li>Does Not Exist The evaluation result is True if the attribute has no value.</li> </ul>
<b>Evaluation</b>	The evaluation result is True or False.

### Example

Build a condition to verify whether the Description attribute is blank.

This table shows the condition:

<b>Attribute</b>	Description
<b>Check</b>	Does Not Exist

The evaluation result of the condition is true if the Description attribute has no value.

## Combined condition

Create a combined condition to evaluate several condition types simultaneously in the monitor rule. All condition types can be combined using the logical operators AND and OR.

The evaluation result of a combined condition is based on the evaluation results of the sub-conditions. Parentheses are used to specify the order of evaluation.

This table shows the syntax:

<b>Syntax</b>	[Condition1] AND   OR [Condition2] You can have more than two sub conditions in one combined condition. For example: [Condition1] AND [Condition2] AND [Condition3]
<b>Condition1</b>	A monitor condition already defined in the list of conditions.
<b>Condition2</b>	A monitor condition already defined in the list of conditions.
<b>Evaluation if AND is used</b>	Evaluates to True if all conditions are True.
<b>Evaluation if OR is used</b>	Evaluates to True if at least one of the conditions is True.

Combined conditions can be nested using this syntax:

<b>Syntax</b>	[Condition1] AND   OR ([Combined Condition])
<b>Condition1</b>	A monitor condition already defined in the list of conditions.
<b>Combined Condition</b>	A monitor combined condition, already built using AND or OR, based on the list of conditions. Combined conditions that are reused are surrounded by round parenthesis.
<b>Evaluation</b>	The comparison conditions are evaluated first. Then the combined conditions marked by parenthesis are evaluated. The evaluation result is True or False.

Note that if one of the sub-conditions is of type `DateTime Check`, the evaluation of the combined condition waits for the timer to expire.

### Example 1

You want to generate an alert if a sales order is shipped, and the shipment is late or it was partially shipped. You define a monitor that evaluates the SalesOrder and Shipment Application Documents.

The SalesOrder and Shipment Application Documents are related. In the monitor definition, you must select a reference between these Application Documents. See [References](#) on page 199.

You create these conditions:

Name	Type	Condition	Description
SalesOrderShipped	Attribute-Value Comparison	<code>SOStatus = Shipped</code>	Checks whether a sales order is shipped.
PartialShipment	Attribute Comparison	<code>ShippedQuantity &lt; OrderedQuantity</code>	Checks whether only a part of the ordered items were shipped.
DelayedShipment	Attribute Comparison	<code>PromisedShipDateT ime &lt; ActualShipd ateTime</code>	Checks whether the shipment was late.

Name	Type	Condition	Description
PartialOrDelayedShipment	Combined Condition	SalesOrderShipped AND (PartialShipment OR DelayedShipment)	Checks whether a partial or delayed shipment occurred.

Define a monitor rule of type "Condition Only". In this rule, select the `PartialOrDelayedShipment` combined condition. If a sales order that has the status *shipped* is intercepted, the monitor waits until it receives the corresponding shipment document. If the sales order and shipment documents are both available. The monitor compares the attributes from the defined conditions. If the combined condition evaluates to true, an Alert is sent.

### Example 2

To generate an alert if a partial shipment occurs for items in the sales order. You define a monitor that evaluates the `SalesOrder` and Shipment Application Documents.

The `SalesOrder` and Shipment Application Documents are related. In the monitor definition, you must select a reference between these Application Documents. See [References](#) on page 199.

Create these conditions:

Name	Type	Condition	Description
SameLine	Attribute Comparison	ShipmentRefLineNr = SOLineNr	Ensures data is read from the shipment that belongs to the order line.
SameItem	Attribute Comparison	ShipmentItemID = SOLineItemID	Ensures shipment data is read for the item used in the order line.
ShipmentOccurs	Attribute-Value Comparison	ShipmentStatus = Shipped	Checks whether items were shipped.
LessQuantity	Attribute Comparison	ShippedQuantity < SOLineQuantity	Checks whether only a part of the ordered items were shipped.
PartialShipment	Combined	ShipmentOccurs AND SameLine AND SameItem AND LessQuantity	Checks whether a partial shipment occurred.

Define a monitor rule of type Condition Only. In this rule, you select the `PartialShipment` combined condition. If a sales order that has the status *shipped* is intercepted. The monitor waits until it receives the corresponding shipment document. If the sales order and shipment documents are both available, the monitor compares the attributes from the defined conditions. If the combined condition evaluates to true, an Alert is sent.

## Advanced properties of connection points

This section describes some advanced properties.

### Application connection points of type IMS

This table shows the advanced properties for application connection points of type IMS:

Category	Property name	Units	Default value	Description
Connection	Parallel Sending	Number of threads	2	Maximum number of threads ION starts in parallel for sending message.

### Advanced properties for polling

For Application Connection Points of type **Infor Application** and **File Connection Points**, this table shows the advanced properties related to the polling service:

Category	Property Name	Units	Default Value	Description
Polling	Sleep Time of Polling Daemon	Milliseconds	5000	The amount of time the polling daemon waits before polling again if no outbox records / files are found during a specified polled instance of time.

Category	Property Name	Units	Default Value	Description
Polling Error Retry	Number of Error Retries	Number	20	For Infor Application connection point. This represents the number of fresh retries the connector makes in succession to retrieve the records from outbox in case of an error. Within each fresh retry, the connector makes a series of attempts determined by the Delay number and the Delay change unit. For File connection Point. This represents the number of retries the connector makes in succession to read files from the configured 'Read Location' in case of an error.
	Default Min. Retry Delay	Milliseconds	10000	Default minimum retry delay in millisecond after error.
	Default Max. Retry Delay	Milliseconds	600000	Within every fresh retry, this is the default maximum retry delay in millisecond after an error. If the attempts due to Default retry delay number and Default retry delay Change unit exceed this value, a new retry attempt is started.
	Default retry delay change unit	Milliseconds	10000	Default retry delay increment time in millisecond after error. This parameter works together with 'Default retry delay number'.
	Default retry delay number	Number	10	Default retry delay number after error within a specified Retry attempt; see "Number of Error Retries". Every successive attempt is delayed cumulatively by the parameter 'Default retry delay change unit'.

This table shows the available additional properties for application connection points of type **Infor Application**:



Category	Property Name	Units	Default Value	Description
Cleanup	Outbox Cleaner Expire Time	Hours	36	Messages older than this time frame, and with status=1 (Message is processed), are cleaned.
	In-box Cleaner Expire Time	Days	30	Messages from the <code>esb_inbound_duplicate</code> table older than this time frame are cleaned regardless of their status.
Connection	Maximum Active Connections	Number	-1	The maximum numbers of connections that can remain idle in the pool, without extra ones being destroyed. A negative number specifies no limit.
	Minimum Idle Connections	Number	20	The minimum number of active connections that can remain idle in the pool, without extra ones being created when the evictor runs. A 0 (zero) creates no active connection.
	Maximum Idle Connections	Number	200	The maximum number of active connections that are allocated from this pool simultaneously. A negative number specifies no limit.

## Protected content

As a Content provider, you can use this function to identify if the Data Flow or Mapping provided by you is not modified in the target system.

Original content can be identified by the **Protected** icon and cannot be directly modified. Protected mapping allows you to approve a protected version and to create a new draft from the protected version. All other modifications, such as duplication, rename, export and import, result in the removal of the **Protected** icon.

To define content as protected on the source system, you must run these tasks:

- Define protected content.
- Export content as protected.

On the target system you must import the protected content.

## Defining protected content

Ensure the `IONContentProvider` role is added to the appropriate users.

You can define protected content for document flows and mappings.

- 1 On the source system, go to either **Connect > Data Flows** or **Connect > Mappings**.
- 2 Click **Protect on Export** on the overview page for one entity from the tile toolbar or enable the multi-select mode and select several entities.  
Alternatively, go to the details page and use the modeler button bar. With this option you can define the selected version of the mapping to be protected on export. Click **Protect on Export**.
- 3 The entity is marked with the **Protected on Export** icon.  
You can reverse the whole action for the document flow or mapping by clicking **Do not protect on export**.

## Exporting content as protected

You can export protected content for document flows and mappings.

- 1 On the target system, select **Connect > Data Flows** or **Connect > Mappings**
- 2 Click **Select** to select one or more entities to be exported.
- 3 Click **Export**.  
When at least one entity marked as **To be Protected** is selected, you can choose to protect the export through the confirmation dialog box. Protect export to create an export file that can be used in other environments. Do not use a protected export to create a backup.  
Do not include properties to create a generic delivery to be used in other environments. For exporting connection properties, click **Yes**.
- 4 The selected entities are exported to the file.

## Importing protected content

You can import protected content for document flows and mappings.

- 1 On the target system, select **Connect > Data Flows** or **Connect > Mappings**
- 2 Click **Import**.  
A dialog box is displayed where you can locate and select the file you created with the export.
- 3 Select the file to import.  
A verification is done when the import file contains entities that are marked as “Protected”. When the file is not modified, all entities from the file are imported and the entities that are marked as “Protected” are Protected. In case a modification is identified, an error message is displayed.

---

## Chapter 5: ION Connect Mapping

A mapping is a description of how to transform (translate) messages, or documents, sent between internal and external resources.

In a homogeneous situation where every component speaks the same language, no mapping is required. But the data from the sending connection point may not match the expectations of the receiving connection points. For example, when using technology connectors such as the Database Connector. In that case you can use a mapping to transform the message contents.

The translated data can be used by other applications connected through ION and enables straight through processing of documents.

ION Mappings provides the capabilities to define mappings between standard or custom documents. A graphical modeler, called the Graphical mapping editor, is used to define the transformation.

You can use a graphical mapping to:

- Change a document. For example map `Sync.MyCustomSalesOrder` to `Sync.SalesOrder`.
- Change the verb of a document. For example, map `Sync.MyProduct` to `Load.MyProduct`.
- Extend the document by adding content or translating data values without changing the document type. For example, map `Sync.SalesOrder` to `Sync.SalesOrder` to add some fixed property values.

You can use a sensitive data mapping to hide certain values or to remove them from the output document. When you use sensitive data mappings, you can perform these actions on document nodes:

- Mask data from nodes.
- Remove data from nodes.
- Remove nodes from a document.

A mapping cannot combine two BODs or split one BOD into multiple BODs. You can achieve the latter with a parallel flow having a specific mapping in each branch.

**Note:** to create mappings, you must understand the structure and contents of the source and target documents. Additionally, for part of the mapping functionality you must also have sufficient background knowledge and experience regarding:

- XML concepts such as elements, attributes and namespaces.
- Programming concepts such as functions, input/output parameters and loops.
- XSLT (Extensible Stylesheet Language Translation).

These topics are discussed:

- Some concepts related to mapping.
- To create, change or delete mappings.

- To use the graphical mapping modeler.
- To use the sensitive data mapping.
- To use functions and user-defined functions.
- To do advanced mapping using XSLT.
- To test a mapping.



An overview of used icons is included.

## Concepts

To use the mapping you must understand several basic concepts.

There are two types of mappings that you can create. The graphical mapping and the sensitive data mapping types.

This table shows the types of mappings:

Icon	Name	Description
	Graphical Mapping	Used for document transformation.
	Sensitive Data Mapping	Used for cleansing sensitive data in a document.

## Graphical mapping

A mapping specifies how the source document is translated to the target document. The source and target documents can either be standard or custom documents. The structure and elements in these documents is defined in the registry.

The mapping consists of a sequence of connections between the elements from the source and the elements from the target document. Additionally, more complex transformations can be added using functions.

**Note:** To ensure correct transformation, root elements always must be matched. Internet Explorer and, to a lesser extent, Edge cannot properly handle the creation of mappings for large documents. Therefore, we recommend that you use the Chrome or Safari browser when viewing or editing mappings.

## Function

A mapping can contain one or more functions. A function transforms the data from the source to the target. For example, it can change the data type of the source element to enable mapping to a target element having a different data type. Alternatively, it can be an advanced function that performs logic and selection operations.

Several standard functions are available. Additionally, a mapping provides the concept of user-defined functions. A user-defined function can be created to perform complex actions of multiple functions within a standard definition that is created using XSLT syntax.

### Standard functions

For details about the standard functions, see [Standard functions for ION mappings](#) on page 510.

## Smart Matching

With Smart Matching you can analyze the source and target document. After analyzing you can map large or complex mappings with automated mappings that are based on preset criteria set by the user. Smart Matching is enabled by default. Smart Matching performs analysis of the source document to find matches based on probability of matches to the target. Smart Matching processes the XSD of the source and target documents into Xpath statements. Matches are made based on the relative XPath and the end element data type throughout the document. You can select the partial probability of a match to be included in the mapping or the full probability. This is used in determining which matches to include within the mapping. Using partial probability includes more matches from outside of the 1:1 matching scope. This is useful where the structure between the documents is different. The partial probability is selected by default. You can select to use the full probability, 100%, to match two documents directly as 1:1 match.

Smart Matching matches are the dashed connection lines between the source and the target documents.

## Smart Matching example

Enable Smart Matching and set **Probability** to **Full**. Only source elements that map directly onto the target are mapped. For example:

- Source document: `Sync.SalesOrder`
- Target document: `Process SalesOrder`

Performing a mapping from these root elements, takes the root element `/SyncSalesOrder` to `/ProcessSalesOrder` as a 100% match. As you defined the relative starting context of the initial mapping point.

The children are then mapped where their XPath matches are 100% and their data types are compatible.

As these BODs are structurally virtually identical with the exception of the verb element. All children which match where the XPath match is 100% and data types are compatible are mapped automatically.

The two child elements that are not mapped on the first pass are `/SyncSalesOrder/DataArea/Sync` from the source and `/ProcessSalesOrder/DataArea/Process` from the target. This is because from the relative context of the starting point of the mapping, the root node level. These XPath statements are not a 100% match.

Performing an additional mapping from the element `/SyncSalesOrder/DataArea/Sync` to `/ProcessSalesOrder/DataArea/Process` sets the relative context of the mapping. It treats the initial mapping point as the set Smart Matching Probability of 100%. The children of this node are mapped. Again, as the structure of these elements within the XSD is virtually identical all children are mapped where there is a 1:1 match. Performing these mappings results in a mapping that contains no error notifications. For example the mapping would produce valid XSLT. There would be warning notifications as all elements in this example are not mapped in the target document. `Process.SalesOrder` contains the attribute `/ProcessSalesOrder/DataArea/Process/@acknowledgeCode` which is not present in the `/SyncSalesOrder` XSD.

To make the mapping warning free and 100% complete, from a target document perspective. You can add a constant value function to the attribute to complete the mapping.

## XSLT Generator V1 and V2

The graphical mapping model is translated to XSLT using an algorithm that generates the code and is available as V1 and V2.

Any new mapping uses by default XSLT Generator V2. The existing XSLT Generator V1 is deprecated but is still used for existing mappings. Mappings that are exported from ION version 12.0.34 and earlier use the XSLT Generator V1. Mappings that are set to use XSLT Generator V1 are in Compatibility Mode. If you open such a mapping, then a warning dialog box is displayed, and a message is displayed in the notification panel.

These are the improvements in V2:

- Generated XSLT code is cleaner, efficient in size and performs faster when executed.
- Removed known limitations.
- Configuration option to use XSLT Generator V1 or V2.
- Auto detecting target parent node occurrence.

When this option is switched on, the occurrence of a non-connected target parent structure is respected when its children are connected.

See [Auto detect parent occurrence](#) on page 138.

These limitations are removed in V2:

- Incorrect output when connecting source nodes of different hierarchical level to a single target node.
- Incorrect output when connecting source nodes and a constant starting point to a single target node.
- Could not assign a Default Value to a connection between source and target element from different hierarchical level.
- Incorrect output, when a multi-occurrence element that was connected to a function input was not defined as type list.

## Compatibility Mode

Compatibility mode can be turned on and off. This depends on if you use XSLT Generator V1 or V2. The configuration is set on a mapping version. You can have multiple versions in a mapping using different XSLT Generator.

To change a mapping version to use the XSLT Generator V2 or V1:

- 1 Select the **Configuration** tab.
- 2 To use V2, switch off **Enable Deprecated XSLT Code Generator (Compatibility Mode)**.  
To use V1, switch on **Enable Deprecated XSLT Code Generator (Compatibility Mode)**.
- 3 Click **YES** to confirm.  
The XSLT code is regenerated for the graphical model.

## Using Compatibility Mode

Specific mapping use cases required to use multiple mappings steps in a document flow, for one document transformation. This was because of the known limitations of XSLT Generator V1.

Keeping those mapping models supported, the **Compatibility Mode** must be switched on. Advanced mapping with hand-written XSLT code must have **Compatibility Mode** switched on, if created with XSLT Generator V1. You can regenerate the graphical model from which the XSLT code was extended.

These are the best practices for changing from V1 to V2:

- Always test the mapping after changing the Compatibility Mode.
- Rework the graphical mapping(s) that used workaround solutions.
- Copy your hand-written XSLT code in a XSLT Generator V2 mapping version. Or rewrite the XSLT code that is generated from a XSLT Generator V2.

We recommend that you change all your mappings from V1 to V2. The XSLT Generator V1 eventually becomes completely deprecated with an announcement in the Release Notes.

## XSLT and advanced mapping

A graphical mapping model is translated to XSLT. XSLT is a language to transform XML documents to other XML documents. From the graphical model, XSLT 2.0 compliant XSLT is created.

An alternative of using the graphical editor is using XSLT directly. This is called advanced mapping. You can use a hand-written XSLT (or an XSLT created using external tooling). Or you can start with the graphical editor and use the generated XSLT as a basis for further editing.

## Sensitive Data Mapping

A sensitive data mapping is used for protection of any known sensitive data transferred in a document.

With the sensitive data mapping type, you can hide and remove data in a mapping step. It is not required to have any XSLT knowledge or to model the transformation in a graphical mapping.

The sensitive data mappings use the same documents as both the Source and the Target documents. The documents nodes are mapped 1:1 apart from the nodes or elements the user selects. For example to mask documents, to remove document data or to completely remove the node and its children from the output document.

## Creating, changing, or deleting mappings

You can create a mapping from the:

- Data flow modeler.
- Mappings list.

### Creating mapping from the data flow modeler

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Connect > Data Flows**.
- 3 Open a data flow or create a new one.
- 4 Add a Mapping activity to the flow.
- 5 In the properties pane, click **New** and select the mapping type.  
The **Mapping** details page is displayed.
- 6 Save the mapping and click **Back** to return to the flow.
- 7 Select **Connect > Mappings**  
A list of existing mappings is displayed including the mapping that was created using the previous steps.

### Creating graphical mapping from the Mappings menu

- 1 Select **Connect > Mappings**.  
A list of existing mappings is displayed.
- 2 Click **Add** and select **Graphical Mapping**.  
The **Mapping** details page is displayed.
- 3 Specify these properties for the new mapping:



**Name**

The name of the new mapping

**Description**

A description of the mapping (optional)

**Source**

The document to map from. Use **Browse** to select a document. The document must be available in the Registry.

**Target**

The document to map to. Click **Browse** to select a document. The document must be available in the Registry.

- 4 Configure the mapping.
  - a Click the **Configuration** tab.
  - b Change these properties if required:
    - Disable Data Type Casting.
    - Disable Allow Mapping Multi-occurrence Elements to Single-occurrence Elements.
    - Disable Allow Mapping Optional Elements to Mandatory Elements.
    - Enable Remove Namespaces from resulting XML
    - Enable Omit Empty Output Tags
    - Disable Smart Matching.
    - Adjust the minimum match probability.
  - c Return to Graphical Mapping from the **Configuration** tab.
- 5 Model a graphical mapping between source and target document.
- 6 Save the mapping.
- 7 Click the **XSLT Edit** tab and review the XSLT that is generated by the graphical mapping.
- 8 Click the **Testing** tab and load a representative XML document.
- 9 Click **Format** to format the input. Test the XSLT that is generated.
- 10 Click **Preview** to test the mapping.
- 11 To format the test result, click **Format**.  
 After a mapping is completed, for example it contains no Error notifications, your mapping is ready for use in a data flow. When you activate the data flow, the ION Service uses the XSLT generated by the mapping. Click **USAGE** to show the list of data flows where the mapping is used.

## Creating a sensitive data mapping from the Mappings menu

- 1 Select **Connect > Mappings**. A list of mappings is displayed.
- 2 Click **Add** and select **Sensitive Data mapping**. The **Mapping** details page is displayed.
- 3 Specify this information:

**Name**

Specify the name of the new mapping.

**Description**

Optionally, specify a description of the mapping.

**Document**

Specify the document from which you want to hide the sensitive data. For sensitive data mappings, the source and target documents are always the same. Click **Browse** to select a document. The document must be available in the Data Catalog

If there is an updated schema for the selected document, a notification message is displayed. Click **Sync** above the document selection field to load the new schema from the Data Catalog.

- 4 Add nodes to the masking table.  
Select a node from the tree that represents the document. Right-click and select **Add to Masking Table** to add the node to the masking table. After a node is added to the masking table, you can apply security policy on it.
- 5 Apply security policies.  
To apply a security policy on an element:
  - a Select the drop-down box in the **Security Policy** column.
  - b Select one of the available security policies:
    - Data removal - removes the data value of an element or attribute in the output BOD.
    - Data masking - masks, that is, replaces, the data value of an element or attribute in the output BOD with a user input string.
    - Node removal - removes the node and its children from the output BOD.
- 6 Save the mapping.
- 7 Click the **Testing** tab and load a representative XML document.
- 8 Click **Format** to format the input.
- 9 Click **Preview** to test the sensitive data mapping. To format the test result, click **Format**.

A mapping is completed, for example, when it contains no error notifications. After a mapping is completed, you can approve that mapping and use it in a data flow.

## Changing a mapping

A mapping can be changed by opening it from the Mappings overview page or from the data flow that uses the mapping.

When a draft version has unsaved changes, an indicator (yellow triangle) is displayed on the draft version in the version list of the mapping.

**Note:**

- Mappings can be reused in multiple data flows. A changed mapping is used when a data flow is activated after the change. Already active data flows still use the older mapping, before the change, until these data flows are re-activated.
- The mapping name and source/target documents can only be changed when a mapping is in “Draft” mode and no pre-existing mappings are approved.

---

## Deleting a mapping

A mapping can be deleted from the Mappings overview page.

If a mapping is used in a data flow, you cannot delete that mapping.

## Using the graphical mapping modeler

You can create a mapping between a source and target document without manually write complex XSLT. You can use a graphical mapping to graphically create a mapping structure, including the use of standard functions without in-depth XSLT knowledge. The graphical mapping is translated into optimized XSLT to be used by the ION runtime.

Graphical mapping consists of connecting nodes, elements and attributes within a source document to a target document. You can do this manually or by utilizing Smart Matching. With Smart Matching, you can select an element from the source and match the element with a compatible element within the target. Automatically the child elements are mapped. Smart Matching is enabled by default. You can disable it by opening the **Configuration** tab.

In the mapping modeler, the left pane shows the elements and attributes of the input document. The elements and attributes of the output document are displayed on the right pane. The central pane shows the definition of the transformation. To change a height of rows of a hierarchy tree, click **Comfortable view/ Compact view**.

## Defining a graphical mapping

- 1 Select the element in the source document tree. The source element is highlighted.
- 2 Move the mouse to the element to be matched in the target document tree. When you move the mouse across the editor window, a dotted red line is displayed. When you pause the mouse on the element in the target document tree to be mapped, it indicates whether the mapping is allowed.
- 3 Click and release the target element to create a mapping. A dotted grey line is drawn between the two elements. If the mapping is not allowed, for example because of type incompatibility, the mouse cursor changes. It indicates the mapping cannot be made to the selected element.

An alternative procedure to define a graphical mapping is:

- a Click and hold the element to be matched in the source document tree.
- b Drag the element from the source document tree to the target document tree. A box containing information about the element is displayed. When you move the mouse across the editor window, a dotted red line is displayed. When you pause the mouse on the element in the target document tree to be mapped, it indicates whether the mapping is allowed.
- c Release the mouse button to create a mapping. A dotted grey line between the two elements is displayed. If the mapping is not allowed, for example because of type incompatibility, the mouse cursor changes. It indicates the mapping cannot be made to the selected element.

**Note:** A solid grey line between the source and target elements, instead of a dotted one, is drawn in these situations:

- A function is added manually.
- Data type casting is applied but smart matching is off.

## Document tree

For each of the source and target documents a document tree is created in the mapping modeler. This is a navigable structure that represents the XSD. Several ways are implemented to view the document information. When a document tree is first displayed, it is open at root node level. The detail for the child elements is collapsed. You can expand the section of interest or the entire document. Use the navigation methods provided.

To reset the default view, click **Reset** to collapse all trees and to remove all filters.

## Modifying the working area of the Mapper window

To extend the working space and show more elements in the Mapper window:

- Click the **Full Window View** icon in the Modeler window. If the Full Window View mode is on, the working area is expanded by hiding the bottom and the left side panels. If this mode is off, these panels are displayed in their normal size.
- Click the **Compact View** icon in the Modeler window. If the Compact View mode is on, more elements are displayed by reducing the height of the rows. If this mode is off, the row height is bigger, but fewer elements are visible.

## Tree navigation

Using the plus and minus icons within the document tree expands or collapse the related element only. If child elements exist beneath the expanded element they remain in a collapsed state. To expand and collapse sections of the document tree is context sensitive. These expand or collapse the entire tree from the selection point down through the document hierarchy. For example, selecting expand on the root node expands the entire document.

## Searching and filtering

You can search and filter elements in the source and target tree.

---

## Searching

To search an element, based on its name, in the source or target tree:

- 1 Click the **Search** icon to expand the search box above the tree.
- 2 In the search box, specify the name or a part of it.
- 3 The search results matching the used filters are displayed. Hover over one of the found nodes to show the full path.
- 4 Click the item you were looking for. This takes you to the corresponding element in the tree.  
You can also use the find next and previous icons to navigate through the search results in the tree.
- 5 Click the filter icon to collapse the search box and open the filters.

## Source tree filtering

You can filter the elements to be displayed in the source tree. To reduce the size of the tree and review only the elements that are used in your mapping, hide the unused elements.

- 1 Click the **Hide all nodes that are used in the Mapping** icon. The filter is activated. Only the elements that are not used in the mapping are displayed.
- 2 Click the **Hide all nodes that are not used in the Mapping** icon. The filter is activated. Only the elements that are used in the mapping are displayed.

To view all elements, unused and used in the mapping, switch both filters off.

**Note:** Parent elements are displayed even though they do not match the filter. For example, this can happen if you select **Hide all nodes that are not used in the Mapping**. In that case, unused elements are still displayed if they have descendants, such as children or grandchildren, that are used.

## Target tree filtering

You can filter the elements to be displayed in the target tree to show the:

- Matched elements to review the mapping that was done so far.
- Elements having warnings, to fix the issues.
- Elements having errors, to fix the issues.
- Elements having functions.
- Elements having default values.
- Elements having comments.

To filter the elements in the target tree:

- 1 Click the **Matched** icon above the tree. If it is switched on, elements that are matched, either directly or through a function, are displayed. If it is switched off, those elements are hidden.
- 2 Click the **Warning** icon above the tree. If it is switched on, optional elements that are not matched or partly matched are displayed. If it is switched off, those elements are hidden.

- 3 Click the **Error** icon above the tree. If it is switched on, elements having errors are displayed. If it is switched off, those elements are hidden.
- 4 Click the **Show only elements having functions** icon above the tree. The **function** filter gets switched on, and elements having functions are displayed. If the **function** icon is switched off, those elements are hidden.
- 5 Click the **Show only nodes having default value** icon above the tree. The default value filter gets switched on, and elements having a default value are displayed. If the default value icon is switched off, those elements are hidden.
- 6 Click the **Show only nodes having comments** icon above the tree. The comment filter gets switched on, and elements having comments are displayed. If the comment icon is switched off, those elements are hidden.

**Note:** Parent elements are displayed even though they do not match the filter. For example, you switch on the **Error** icon and switch off the **Matched** and **Warning** icons. In that case, an element with no errors can still be displayed if it has a descendant, such as a child or a grandchild.

## Refreshing

You can refresh the source and the target documents of the existing mapping to get the latest schema updates. For example, elements were added or removed from the custom BOD.

Click the **Sync** icon above the source or the target tree to get the latest metadata for each document.

When either the source or target document are not registered in the Data Catalog a warning is displayed and refreshing is disabled. After the documents get registered in the Data Catalog you can refresh with the newest schema.

When refreshing the schema, any removed or modified elements are displayed. Click the **Export** icon to export the elements to CSV.

Limitations in current version

Changes to the `UserArea` extensions are not detected or indicated. If you notice that `UserArea` extensions are added or changed for a document, you can click the **Sync** icon to refresh the trees with the latest schema and `UserArea` extension.

## Focus view

The mapping modeler has various visual indicators to simplify the mapping view for the user. Focus view shows information relevant to the areas of the mapping you are working on to simplify the mapping process.

---

## Graphical mapping connections

Mapping connections at the node level are displayed within the visible working area. Element and attribute mapping connections are displayed when a mapped element or attribute is selected within the document tree.

A mapped element or attribute may get beyond the visible page area by scrolling through either the source or the target document tree. In that case, the mapping connection is no longer displayed, but remains present in the mapping.

## Commenting on graphical connections

Comments can be added on individual graphical connections and on functions where the starting point is a constant.

In both cases, this is visualized as a line to a target node, either coming from a source node, or coming from a function icon.

The comment is inserted as a comment tag in the XSLT code that the graphical mapping is generating. To view the comments in the XSLT code, open the XSLT Editor tab. Comments in the XSLT code additionally show the author of the comment and the date and time when it was added.

Comments can be added, edited and deleted from:

- Graphical connection lines in the graphical mapping modeler.
- Function scope container, in the Function Editor.

When two connection lines are overlapping, you can select the Source Path of the connection from the comments dialog box. Then you can add, edit or delete a comment. This scenario happens when the target is connected to a source node and constant starting point. Or when there is more than one constant starting point that is connected to a single target node.

### Adding a comment

- 1 Right-click an existing graphical connection line and select **Add Comment**.  
The **Add Mapping Comment** dialog box is displayed.
- 2 Specify your comment in the text area. The length of the comment is limited to 255 characters.
- 3 Click **OK** to save the comment. Alternatively, click **CANCEL** to discard the changes and exit.  
The comment icon indicates the newly added comment on the connection.

### Editing a comment

To edit a comment:

- 1 Right-click an existing graphical connection line that contains a comment.
- 2 Select **Edit Comment**.

The **Edit Mapping Comment** dialog box is displayed with the contents of the comment.

- 3 Edit the comment and click **OK** to save the comment.

If you delete the contents of the comment and click **OK**, the comment is removed.

## Deleting a comment

You can remove an existing comment in one of these ways:

- Right-click the graphical connection and select **Delete Comment**.
- When editing a comment, clear the contents of the comment input field and click **OK**.

The comment icon is removed after you delete the comment for the connection.

**Note:** If there is no existing comment, the **Delete Comment** function stays hidden. Each container can contain only one comment.

## Comments on function scope container

You can add, edit and remove a comment from within the function editor.

Right-click a function scope container and select one of these options:

- Add: To add a new comment.
- Edit: To edit an existing comment.
- Delete: To delete an existing comment.

Comments placed through the function editor are also shown on the respective connection lines or constant starting points when using the graphical mapping modeler.

## Visualizing an off screen mapping connection

- 1 Select the matched element from the target tree.
- 2 Right-click the element and select **Focus**.

The source document tree starts scrolling and brings the mapping connection back into the working area. All mapping connections have a visual icon on the target document tree. It shows its mapping status. This table shows the connection icons:

Icon	Description
Green circle with tick mark	A node, element or attribute is completely matched
Green circle	A node or element that is matched by a function on a parent node. The applied function returns an 'anyType' output. The function is providing all the data for the children nodes/elements/attributes, causing green circle badges on all of the children.
Yellow triangle	Unmatched optional node, element or attribute.



Icon	Description
Yellow triangle with exclamation mark	<p>Warning – a non-mandatory node, element or attribute is partially matched.</p> <p>Warning – an element or attribute is overloaded: it has two or more matches from the source document.</p> <p>Warning – a node is not explicitly matched, but a link is inferred from the connection of one or more child elements.</p>
Red circle with exclamation mark	Error – a mandatory node, element or attribute is not completely matched.

For more information see "Icons and visual feedback".

## Visualizing XPath

- 1 Select the matched element from the source or the target tree.
- 2 Right-click the element and select **Show XPath**. The modal window containing the full XPath is displayed. If the XPath is too long, a horizontal scroll bar is enabled to handle the content of the modal window.
- 3 Click **OK** to close the modal window.

**Note:** The same visualization feature is implemented in the Function Editor.

## Set default value

Use this function to assign a specific default value to a target element. The value is applied only if the corresponding mapped source element is not included.

### Limitations in the current version

The default value can be used only if these conditions are fulfilled:

- The source-to-target mapping connection exists.
- The mapping connects target elements on the same hierarchical level. Default value addition is not allowed for connections between target elements on different hierarchical levels in the source and target trees respectively.
- One-to-many mapping cardinality is allowed.

## Setting the default value

To set the default value:

- 1 Select the required matched target element from the target tree.
- 2 Right-click the element and select **Set Default value**.

A dialog box containing an input field for the default value is displayed. The default value can be a string or a number.

- 3 Specify the value and select the type.
- 4 Click **OK** to confirm the default value or **Cancel** to discard it.

## Deleting the default value

- 1 Right-click the target element containing the default value..
- 2 Select **Delete Default Value**  
The Deletion confirmation dialog box is displayed with a preview of the default value.
- 3 Click **OK** to confirm and delete the Default Value.

## Deleting the default value during edit

- 1 Right-click the target element containing the default value.
- 2 Select **Edit Default Value**.
- 3 Delete the content of the input box.
- 4 Click **OK**.  
The Deletion confirmation dialog box is displayed with a preview of the default value.
- 5 Click **OK** to confirm and delete the Default Value.

## Grouping

Sometimes the source document is a flattened, denormalized, document, while the target document is a structured, normalized, document. This occurs for example if the source document comes from a fixed-length or delimited file as picked up by the file connection point. In that case, you can use the 'group by' functionality.

The example shows that without grouping, the order header is repeated in the target document like in the source document. When using grouping, the two lines for order SO1 are combined in a single SalesOrder document having a single header.

This is the code of the source document:

```
<MyOrder>
  <OrderNr>SO1</OrderNr>
  <Status>Open</Status>
  <LineNr>1</LineNr>
  <Item>ITM11</Item>
  <Quantity>110</Quantity>
</MyOrder>
<MyOrder>
```

```
<OrderNr>S01</OrderNr>
<Status>Open</Status>
<LineNr>2</LineNr>
<Item>ITM12</Item>
<Quantity>120</Quantity>
</MyOrder>
<MyOrder>
<OrderNr>S02</OrderNr>
<Status>Approved</Status>
<LineNr>1</LineNr>
<Item>ITM21</Item>
<Quantity>210</Quantity>
</MyOrder>
```

This is the code of the resulting document if grouping is not used:

```
<SalesOrder>
<SalesOrderHeader>
<DocumentID>
<ID>S01</ID>
</DocumentID>
<Status>
<Code>Open</Code>
</Status>
</SalesOrderHeader>
<SalesOrderLine>
<LineNumber>11</LineNumber>
<Item>
<ItemID>
<ID>ITM11</ID>
</ItemID>
</Item>
<Quantity>110</Quantity>
</SalesOrderLine>
</SalesOrder>
<SalesOrder>
<SalesOrderHeader>
<DocumentID>
<ID>S01</ID>
</DocumentID>
<Status>
<Code>Open</Code>
</Status>
</SalesOrderHeader>
<SalesOrderLine>
<LineNumber>12</LineNumber>
<Item>
<ItemID>
<ID>ITM12</ID>
</ItemID>
</Item>
<Quantity>120</Quantity>
</SalesOrderLine>
</SalesOrder>
</SalesOrder>
```

```
<SalesOrderHeader>
  <DocumentID>
    <ID>SO2</ID>
  </DocumentID>
  <Status>
    <Code>Approved</Code>
  </Status>
</SalesOrderHeader>
<SalesOrderLine>
  <LineNumber>21</LineNumber>
  <Item>
    <ItemID>
      <ID>ITM21</ID>
    </ItemID>
  </Item>
  <Quantity>210</Quantity>
</SalesOrderLine>
</SalesOrder>
```

This is the code of the resulting document if grouping is used:

```
<SalesOrder>
  <SalesOrderHeader>
    <DocumentID>
      <ID>SO1</ID>
    </DocumentID>
    <Status>
      <Code>Open</Code>
    </Status>
  </SalesOrderHeader>
  <SalesOrderLine>
    <LineNumber>11</LineNumber>
    <Item>
      <ItemID>
        <ID>ITM11</ID>
      </ItemID>
    </Item>
    <Quantity>110</Quantity>
  </SalesOrderLine>
  <SalesOrderLine>
    <LineNumber>12</LineNumber>
    <Item>
      <ItemID>
        <ID>ITM12</ID>
      </ItemID>
    </Item>
    <Quantity>120</Quantity>
  </SalesOrderLine>
</SalesOrder>
<SalesOrder>
  <SalesOrderHeader>
    <DocumentID>
      <ID>SO2</ID>
    </DocumentID>
    <Status>
```

```
<Code>Approved</Code>
</Status>
</SalesOrderHeader>
<SalesOrderLine>
  <LineNumber>21</LineNumber>
  <Item>
    <ItemID>
      <ID>ITM21</ID>
    </ItemID>
  </Item>
  <Quantity>210</Quantity>
</SalesOrderLine>
</SalesOrder>
```

To switch on grouping:

- 1 In the source tree, right-click the element that contains the data to be grouped. In the example above, that would be the `MyOrder` element.
- 2 Select **Add group by**. The element gets a visual indication that the grouping is modeled.

To switch off grouping:

- 1 In the source tree, right-click the element that has the grouping indicator.
- 2 Select **Remove group by**.

The grouping works as follows:

These child elements of the 'group by' element, `MyOrder`, are used for the grouping child elements that are connected to:

- Direct children of the corresponding element in the target document, `SalesOrder`.
- Children of non-repeating nodes such as `SalesOrderHeader` inside that target element.

In the example that would be `OrderNr` and `Status`.

If, at runtime, the original XML document has multiple `MyOrder` instances that have the same value for all elements used in grouping. These elements result in a single `SalesOrder` in the resulting XML document. For example, if the `OrderNr` is the same, but the `Status` differs, it results in two `SalesOrder` nodes.

The child elements of the 'group by' element, `MyOrder`. That are connected to a child of a repeating node, such as `SalesOrderLine`, inside the corresponding target element, `SalesOrder`. Are not used for the grouping.

## Error and warning notifications

When performing mapping connections, notifications that are based on your actions are created and displayed simultaneously. Mapping notifications are displayed within a collapsible panel at the bottom of the mapping modeler.

By default, the bottom panel shows only the Mapping Notifications header, and the paginator is hidden. If the bottom panel is expanded upwards, the paginator is displayed.

The summary counts show the status for the:

- Number of errors in the mapping
- Number of warnings in the mapping

Error and warning notification details are displayed in severity order to assist you in completing the mapping. Notifications are displayed in context to where you are working within the target document tree. Select the root node to display all the errors and warnings for the entire document map. To update the notifications select a child node or element.

Each error or warning that is shown in the notifications panel contains information and a link. This link goes to where the notification is relevant within the target document tree. Click one of these notifications to select the corresponding element or attribute in the target document tree. The element or attribute is focused to the center of the mapping modeler working area.

## Exclude from mapping

Performing a mapping manually or with the Smart Matching on, can overload some elements in the target. Overload means an element in the target document has more than one mapping. Sometimes, this can be intentional. For example you can use a function to concatenate two or more elements from a source document into a target element.

In other circumstances this can be unintentional. For example, if Smart Matching is set with a low minimum probability, and a source document contained many instances of a particular named element. Smart Matching can suggest multiple possible matches from the source for a single element in the target document. If the document structures are similar, identify the incorrect connection and delete it from the mapping. If the documents have different structures it can be unsure which is the correct mapping to make. You can exclude a mapping from being part of the resultant XSLT without removing it from the graphical mapping model. You can test the XSLT by excluding the overloading elements in turn to ensure the correct one is used in the final document mapping.

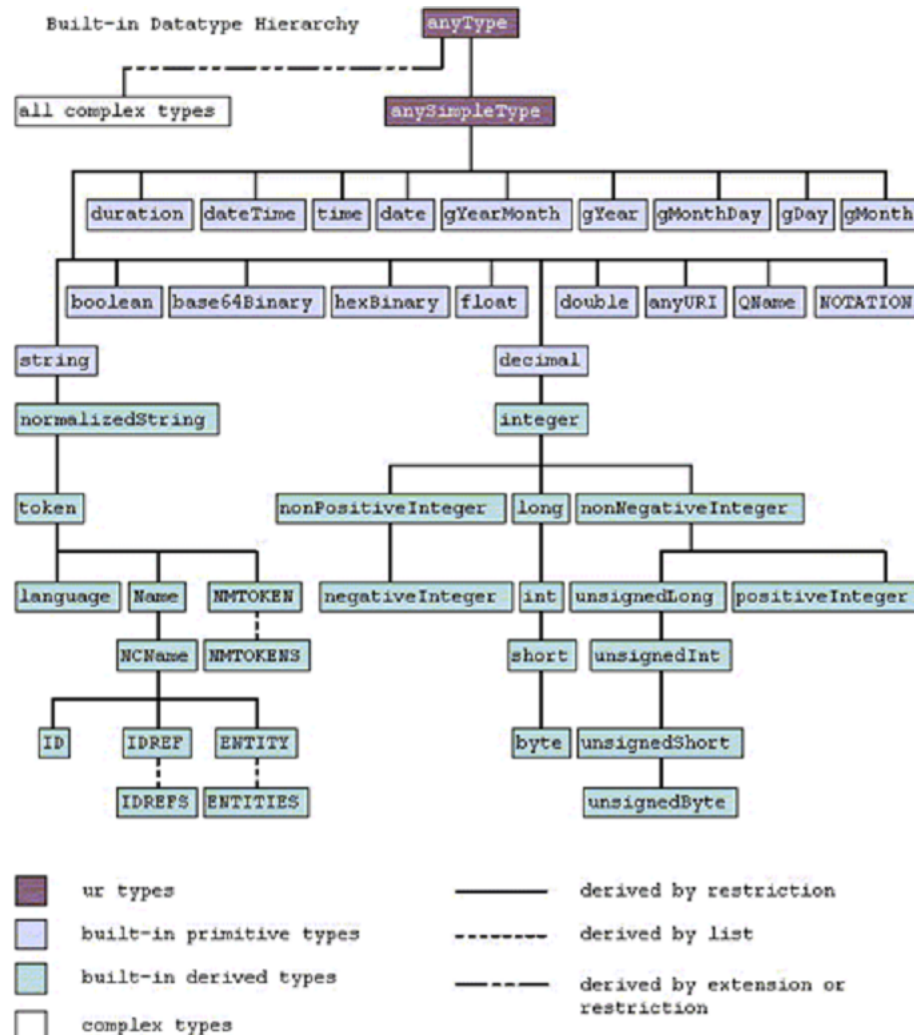
## Excluding an element from a mapping

- 1 Click the overloaded element in the target tree.  
Notifications relevant to the selected element are displayed on the notifications panel.
- 2 Select the element to exclude in the notification panel.
- 3 Select the **Exclude from mapping** check box.

The mapping connection is hidden in the graphical mapping. Graphical mapping icons and notification counts for the mapping are updated. The notification message for the exclusion remains in the notifications list and can be cleared and changed as required.

## Graphical mapping data type compatibility

Data type compatibility checking is implemented, which is done when performing a mapping. Elements from the source are only permitted to be mapped directly to a compatible element type in the target. If a mapping is not compatible the editor does not approve a mapping connection. If the data type of an already mapped source or target element is changed in the schema and the new data types are not compatible. The mapping connection is removed. The mouse icon changes to indicate an incompatible type. The W3C website (w3.org) diagram shows the type compatibility:



**Note:** When using a function that returns an output that is of `anyType`, you can define the target and its children. No validation error is raised if the element with such function has one or more required children when they are not matched. Check whether the required child nodes in the target are set correctly, either by the aforementioned parent function or in separate connection with the source.

To map between incompatible types one or more functions are required. Add them to the mapping between the source and the target elements of the mapping. You can perform casting manually by

creating a function. You can also automatically cast from the source type to the target type where a casting function is available.

## Automatic casting

If automatic casting is on, a data type casting function is applied automatically in these situations if:

- You connect two elements having incompatible data types in the mapping modeler.
- You do the same in the function editor.
- Two elements having incompatible data types are connected automatically because of Smart Matching.

If automatic casting is off, you cannot connect two elements having incompatible data types.

To enable automatic casting between incompatible data types:

- 1 Click **Configuration**.
- 2 Enable **Data Type Casting**.

`anySimpleType` used in template function `get-first-value`, to complete the function connections.

To manually create functions, see "Graphical mapping using functions".

## Element occurrence compatibility

In the definition of a document as stored in the Data Catalog, the occurrence of elements is specified:

- An element can be single-occurrence it occurs once, or an element is multi-occurrence and is repeated multiple times. For example, an order header occurs once for each order document, an order line can occur multiple times in a single order document.
- An element can be optional. It does not have to be included, or an element is mandatory and is required if its parent element exists. For example, inside the `DocumentID` element, the `ID` child element is mandatory, while the `RevisionID` is optional.

The occurrence of an element is displayed behind the element name in the source and target trees. For example:

- `[0..1]` means the element is optional, indicated by the '0', and single-occurrence, indicated by the '1'.
- `[1..∞]` means the element is mandatory, indicated by the '1', and multi-occurrence, indicated by the  $\infty$  sign.
- If nothing is displayed behind the element name, the element occurs exactly once.

This table shows to what extent you can map elements having a difference in occurrence.

Source element	Target Element	Notes
Single occurrence	Multi-occurrence	This mapping can be done.



Source element	Target Element	Notes
Multi-occurrence	Single occurrence	<p>If <b>Allow Mapping Multi-Occurrence Elements to Single-Occurrence Elements</b> is off, this mapping cannot be done.</p> <p>If <b>Allow Mapping Multi-Occurrence Elements to Single-Occurrence Elements</b> is on, this mapping can be done. In that case, and the source document contains multiple occurrences of the element, only the first occurrence is used. The others are ignored.</p>
Mandatory	Optional	This mapping can be done.
Optional	Mandatory	<p>If <b>Allow Mapping Optional Elements to Mandatory Elements</b> is off, this mapping cannot be done.</p> <p>If <b>Allow Mapping Optional Elements to Mandatory Elements</b> is on, this mapping can be done.</p> <p><b>Note:</b> If the source document does not contain the element, the target document also does not contain the element. In that case the resulting document does not match the definition. Use this option only if you are sure that the documents for which the mapping is used contain the source element. This is not uncommon, because in standard BOD definitions most elements are optional.</p>

## Allowing mapping multi-occurrence elements to single-occurrence elements

- 1 Click the **Configuration** tab.
- 2 Enable **Allow Mapping Multi-Occurrence Elements to Single-Occurrence Elements**.

## Allowing mapping optional elements to mandatory elements

- 1 Click the **Configuration** tab.
- 2 Enable **Allow Mapping Optional Elements to Mandatory Elements**.

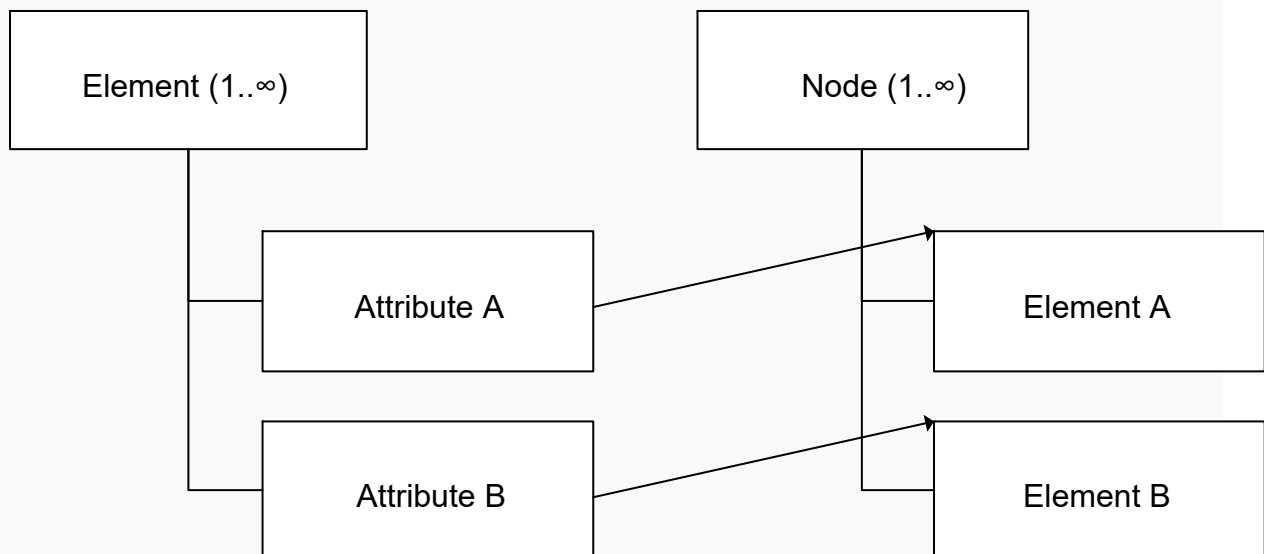
### Auto detect parent occurrence

You can switch on auto detecting the occurrence for target nodes which are not connected, and their children are connected. The connected children are displayed in the output with the respective occurrence of their parent node.

Example: Connecting the attributes of a multi-occurrence source element to the child elements of a multi-occurrence target node.

The target node and elements are expected to be repeated in the output for each occurrence of the source.

To output the expected structure for the target document, the source element and target node must be connected. But that kind of connection is not compatible and is not allowed in the graphical modeler because of their incompatible types.



Even though there is no explicit connection between the parent element and the node. The target node and its children are repeated in the output according to the schema definition if the **Auto Detect Parent Occurrence** is switched on.

---

## Enabling auto-detect parent node occurrence

You can enable auto-detect parent node occurrence, for a mapping version:

- 1 Click the **Configuration** tab.
- 2 Ensure **Compatibility Mode** is switched off.  
See [XSLT Generator V1 and V2](#) on page 118
- 3 Switch the **Auto Detect Parent Occurrence** on.
- 4 Test the mapping.

## Graphical mapping using functions

To perform data transformation you can add a function to complete a mapping.

- 1 Right-click the element in the target document tree where the function is required.
- 2 Click **Add Function**.

The mapping functions editor is loaded and split into two areas. On the left side, the available functions are shown. You can add them to the function modeling editor. On the right side, the function modeling editor is shown. You can connect the functions to the source and target documents.

On the **mapping function editor** window, each mapping context created is represented by a resizable mapping scope container. If there is no container a new context is not created. You can create a new context in these ways:

- Dragging a source element from the source document tree.
- Use the context (right-click) menu to create a constant starting point.

Alternatively, you can add a user-defined function from the editor's left panel. Select **Template Functions > create-custom**.

When a user-defined function is defined as a starting point it is created without any input parameters. A single mapping to the target is shown as a single mapping container with a grey dashed line from the source to the target. It represents the direct mapping between the source element and target element with no function.

Where multiple mappings exist to the target, multiple mapping containers are displayed containing the individual scope of the mapping. You can add one or multiple functions to each scope of graphical mapping. Or merge the individual scopes into a single scope. For example if mapping multiple source elements to be concatenated into a single element in the target.

- 3 Create a single function scope.

Drag the function scope starting point from its mapping scope container into the container where the scope is to be merged. A function scope starting point is a source element indicated by the lock icon. The container where the scope is to be merged is a source element. The scope being dragged into becomes the starting point for the scope of the merged function.

When a function is added to any mapping scope the direct connection, grey dashed line, is removed.

A created mapping within the graphical mapping view shows the selected target element and each of the sources mapped with it. A lock symbol indicates that you cannot remove the target element and each of the sources from the function modeling editor view.

You can view the XSLT code that is generated for the function scopes by clicking the **XSLT Viewer** tab. If a function is invalid, the **XSLT Viewer** tab is disabled, and the XSLT code is not generated. Invalid functions cannot be saved.

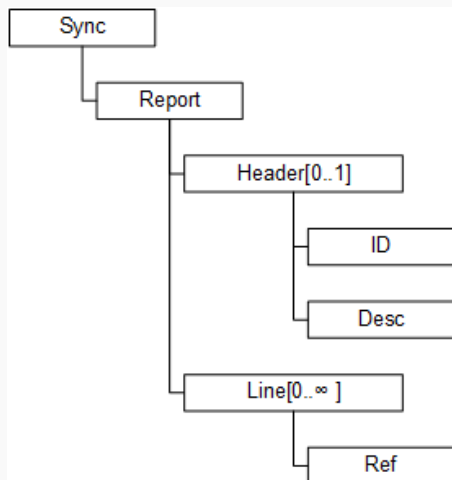
## Starting points of functions

When an element is mapped from source to target it is relative to another element. The relative element can be itself, a sibling, a parent or any descendant of a sibling. The starting point defines the reference from which the subsequent mapping is defined.

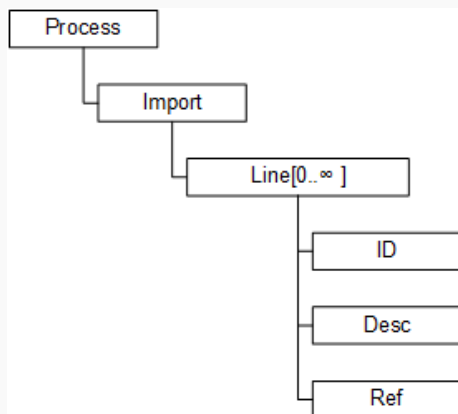
### Example:

Consider a transformation between two documents. The source document is a report that contains header and details. The target document is a flat transformation where the header and details are amalgamated.

The source document schema is represented as a single header with multiple lines (details):



The target document schema is represented as a single import with multiple lines (details):

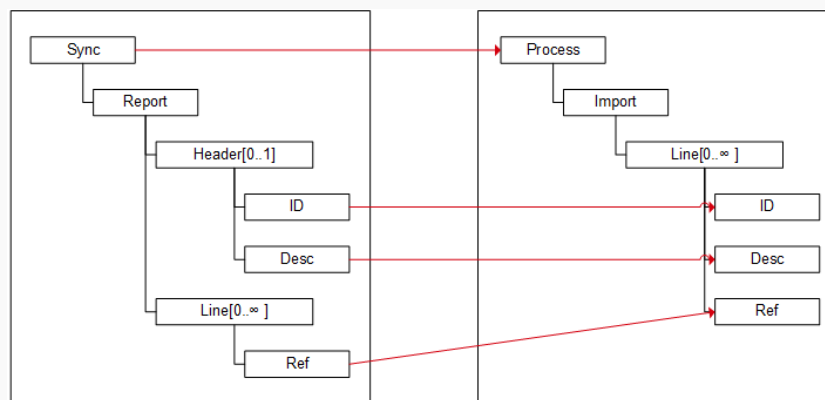


The mapping between the two documents is defined as:

`\Sync\Report\Header\ID` is mapped to `\Process\Import\Line\ID`

`\Sync\Report\Header\Desc` is mapped to `\Process\Import\Line\Desc`

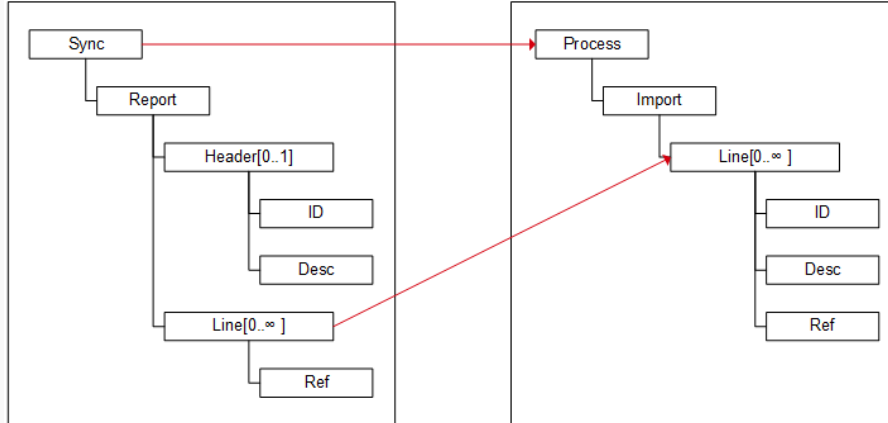
`\Sync\Report\Line\Ref` is mapped to `\Process\Import\Line\Ref`



Mapper optimizer does not know that the Header and Line elements are to be amalgamated. It attempts to create a `\Process\Import\Line` entry for each `\Sync\Report\Header` and each `\Sync\Report\Line`. Consequently, this results with the incorrect XSLT.

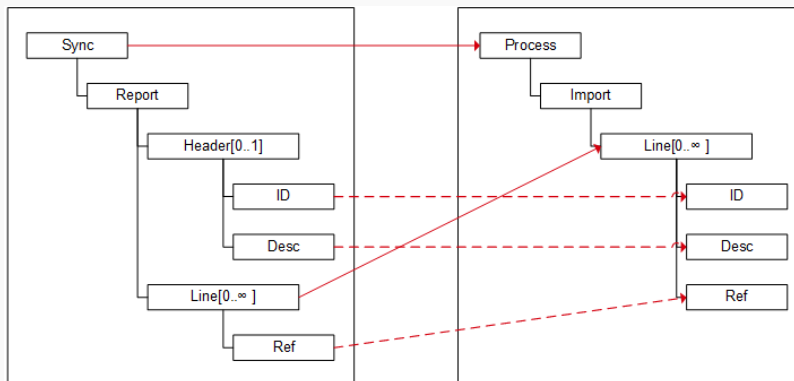
When elements are mapped it is imperative to understand why that mapping is taking place and to determine a correct starting point. That is, for the source document, the node `\Sync\Report\Line` is the starting point as the mapping results with a document containing all line details.

The header information is to be repeated for each line; thus, the `\Sync\Report\Header` elements are relative to `\Sync\Report\Line`.



The `\Sync\Report\Header\ID` and `\Sync\Report\Header\Desc` elements must reference the relative starting point `\Sync\Report\Line`

To achieve this use the function modeler and create a function that connects the source element to the respective target element. Select the correct starting point within the source element widget from the drop-down list.



## Using function scopes

When working with multiple scope functions you can collapse the details of individual mapping function scope containers. The workspace is maximized to focus on a single mapping scope.

- 1 To collapse a function scope container:
  - a Right-click inside the mapping function scope container.
  - b Select **Collapse**.

The mapping scope container collapses to a minimized view representing the detail of the mapping scope. The collapsed view shows a title representing the source of the mapping. If complete a tick icon is displayed in the bottom right corner of the minimized view. If the mapping function is incomplete a warning is displayed in place of the tick icon.

- 2 To expand a function scope container:

- a Right-click inside the collapsed mapping function scope container.
  - b Select **Expand**.
- 3 To add a function to the function scope:
  - a Click and hold the function.
  - b Drag the function from the function tree to the function modeler editor to the relevant function scope. A widget representing the function is displayed.
  - c Release the function. A function is created on the editor window.
- 4 Click **Check mark** to keep your function and close the Function Editor window.

**Note:** If your function is invalid, a confirmation window is displayed.

Click **Cross** to close the Function Editor window. Your edits are discarded.

## Adding constants or user-defined functions

You can add these functions directly in the function editor 's scope container:

- Constants
- User-defined functions

To add either of these functions:

- 1 Right-click in the mapping scope container within the function editor.
- 2 Select **Add Constant** or **Add User-Defined Function**.

Alternatively, you can add a constant or a user-defined function by dragging a widget from the editor's left panel. To add a constant, use **Constant > String or Number**. To add a user-defined function, use **Template Functions > create-custom**.

**Note:** You can also right-click the blank canvas and select **Add Constant**. A separate scope container is created with the constant as a starting point.
- 3 Click **Check mark** to store your changes to the function and close the Function Editor window.

**Note:** If your function is invalid, a warning is displayed.

Click **Cross** to close the Function Editor window. Your changes are discarded.

## Constants

Constants can contain a string or a number value that is passed on as an input to functions, or to a target node.

Constants can be a starting point of a function scope container. This is indicated by the lock sign on the constant widget. A constant which is a starting point cannot be removed from the function container. In such a case, delete the whole function container.

In scenarios where you have a Default Value and a Constant assigned to a single target node, the Default Value is not taken into consideration.

For more details on Default Value, see [Set default value](#) on page 129.

## Adding a constant

To add a constant

- 1 Select **Constant > String or Number**.
- 2 Drag the widget into the scope container.  
You can also right-click a scope container and select **Add Constant**.
- 3 Specify the value of the constant.
- 4 Click **OK** to place the constant.

## Adding a constant as a starting point

To add a constant, that is the starting point of a function scope:

- 1 Select **Constant > String or Number**.
- 2 Drag the widget into a blank canvas.  
You can also right-click a blank canvas and select **Add Constant**.
- 3 Click **OK** to place the constant.  
This creates a new function scope container where the constant is the starting point. As the constant is a starting point, it cannot be deleted. To remove the starting point, you must delete the whole function scope container by right-clicking the function container and selecting **Delete**.

## Editing a constant

Constants that are already added to a function scope container can be edited. You can change the data type of the constant and you can change the value of the constant.

To edit a constant:

- 1 Right-click the Constant widget.
- 2 Select **Edit**.
- 3 Specify the new value in the input field.  
You can change the data type if required.
- 4 Click **OK** to confirm the changes.

## Deleting a constant

Depending if a constant is a starting point of a function scope, you can either remove just the constant, or delete the whole function scope container.

### Deleting a constant that is not the starting point

To remove a constant, that is not the starting point of a function scope:

- 1 Right-click the Constant widget.



- 2 Select **Delete**.
- 3 Click **Yes** to confirm the removal.

### Deleting a constant that is the starting point

To remove a constant, that is the starting point of a function scope:

- 1 Right-click the function scope container.
- 2 Select **Delete Container**.
- 3 Click **Yes** to confirm the removal of the container.

**Note:** Removing a container also deletes any widgets placed in the container.

## User-defined functions

User-defined functions are a special type of function. You can create complex functionality within a single function using XSLT 2.0 syntax and maintain the visual mapping. You can reuse saved user-defined functions in multiple mappings through import.

### Creating a user-defined function

When creating a user-defined function, you must specify the UDF name, input and output parameters, and the XSLT code to make the UDF valid.

To add a user-defined function:

- 1 Select **Template Functions > create-custom** and drag the widget into the scope container. Alternatively, right-click in the function editor's scope container and select **Add User-Defined Function**.
- 2 Specify these user-defined function properties:

#### **Function name**

Specify a unique identifier for the function being created.

#### **Input Parameter Name and dataType**

Specify the name and data type of the input parameters of the function. The selected input data type of each property must match the types of the function inputs.

To add more input parameters, click the **+** icon.

#### **Output dataType**

Specify the type of the output of the user-defined function. The selected output data type must match the type of the next function or the type of the target element.

#### **Namespace <ns>**

The declared function must come from a namespace outside of the XSLT namespace. The URL that is provided does not have to exist but you must associate it with your company and project. For example, `http://yourcompanyandproject`.

The **NameSpace <ns>** field is located in the pop-up window that is displayed when you click the **Edit Function** icon. This icon is located in the lower left corner of the user-defined function widget.

**Function Body**

Declare the function in XSLT 2.0 syntax.

The **Function Body** field is located in the pop-up window that is displayed when you click the **Edit Function** icon. This icon is located in the lower left corner of the user-defined function widget.

**Parameter as dataType**

Specify input parameters for the function. The data type must match the output data type of the function input.

Click the **+** icon to add input parameters.

**Return dataType**

Specify the output type of the user-defined function. This must match the input type of the next function or the end node.

**Omit parent elements**

Select this check box to omit parent XML elements. The parent elements are not transferred to the XML output.

**User-defined function example**

This example shows how to create a user-defined function. Two input parameters of type decimal and a return type of type decimal are used. You can create a function to calculate a percentage that is based on the value of the two input parameters.

Create a user-defined function:

- 1 Specify this information:

**NameSpace <ns>**

`http://inforTemplateFunctions`

**Function name**

Percentage

**Parameter as dataType**

value1 - decimal

value2 - decimal

**Function Body**

```
<xsl:value-of select="(100* $value2) div $value1" />
```

**Return dataType**

decimal

- 2 When you have a fully defined user-defined function, you can save it and reuse it within the scope of the current mapping version.

The user-defined function is saved to the Function Tree under User-Defined Functions.

---

## Editing a user-defined function

The changes that you make by editing a user-defined function are reflected to any function that is using the UDF.

- 1 Click the **Edit** icon on the user-defined function widget.
- 2 Edit the input parameters, output parameter, and XSLT code.
- 3 Click **UPDATE**.

If the UDF was used in other functions, a confirmation dialog box is displayed that indicates the validity of the UDF usage.

- An error icon is displayed if the changes in the UDF have made a function invalid.
- A confirm icon is displayed if the changes in the UDF have not affected the function and did not make it invalid.

- 4 To confirm to update the UDF, click **YES** in the confirmation dialog box.

Functions that are invalid because of a UDF update are indicated in the notification panel in the graphical modeling view. You must correct those functions before you approve the mapping.

## Duplicating a user-defined function

You can create a copy of a user-defined function into the current function scope container where the UDF is used.

- 1 Right-click a user-defined function widget and select **Duplicate**.  
A nameless UDF widget is created, with the configuration of the UDF you duplicated.
- 2 Click **UPDATE** to save the changes in the new UDF.

## Template Functions

Template functions are pre-defined User-Defined Functions. These functions leverage the capability of User-Defined Functions but are specific reusable functions defined by Infor. Template functions can be found in the Function tree under the Template functions heading. They can be used in the same way as any standard function from the function tree. Template functions look like User |Defined Functions, but are in a read-only state and cannot be edited.

## Global user-defined functions

A global user-defined function is a system wide UDF. A UDF that is saved in a mapping version can be declared global. This way, you can make the UDF available for other users, or reuse it in other graphical mappings. If you use a global UDF in a function, a UDF copy of it is created, that must be saved for the mapping version.

A Global UDF may have a description and may also show when and by whom it was updated.

### Creating a global UDF

Creating a global UDF is conditioned on the existence of a previously saved user-defined function.

To create a global UDF:

- 1 Right-click a user-defined function and select **Make Global**.
- 2 Specify a name for the global UDF.
  - By default, the UDF name is used.
  - UDF and Global UDF can have the same name.
- 3 Click **OK** to confirm.

The newly created global UDF is displayed under Global User-Defined in the Functions list.

### Using a global UDF

- 1 Click and drag the global UDF to add it to the function's scope container.  
A local UDF copy is placed in the container.
- 2 The UDF is in edit mode. Specify a unique UDF name.
- 3 Continue to model the function by connecting the UDF input and output parameters

### Updating a global UDF

You can update a Global UDF by overwriting it with a previously saved user-defined function.

- 1 Right-click a UDF name and select **Make Global**.
- 2 Specify an existing name of a global UDF.  
An info message is displayed if you are overwriting.
- 3 Click **OK** to confirm.

### Deleting a global UDF

- 1 Right-click the global UDF and select **Delete**.
- 2 Click **YES** to confirm.

### Adding a description to a global UDF

- 1 Right-click the global UDF and select **Add Description**.  
If a description already exists, you can edit or delete it by selecting **Edit Description** or **Delete Description**.  
A dialog box is displayed where you can change the description.
- 2 Click **OK** to confirm.

### Viewing details of a global UDF

- 1 Right-click the global UDF.
- 2 Select **Details**.

A dialog box is displayed with the description, last updated date, and author.

## Connecting functions

When the sequence of function elements are executed, data is transformed and transferred from the source document to the target document. The source and target of this data transfer is defined by connecting function elements.

To complete a function ensure that it is connected to the source and target elements. Inputs to any function modeling elements must be from the output of another function or a node, element or attribute within the function scope on the function modeling editor. Outputs from any function modeling elements must go to the input of another function or a node, element or attribute within the same function scope.

Source output to Function input.

- 1 Click and hold the source widget.
- 2 Drag the widget to the function input.
- 3 Release the widget.

## Function output to target element or function input

- 1 Click and hold the function widget `data` type.
- 2 Drag the widget to the Target input element name or the function input of the function widget.
- 3 Release the widget.

A dashed red connection line between the two widgets is displayed, when the function model connection is successful. When the function model is complete, the red dashed lines changes to a solid grey line.

## Adding and removing function inputs and outputs

You can configure functions such as Substring and Concatenate to have additional input parameters.

To add or remove parameters:

- 1 Click the **+** icon or right-click the function widget.
- 2 Select **Add Input** to add an additional input parameter to the function.
- 3 To remove an input parameter that is no longer required in the function:
  - a Right-click the function parameter.
  - b Click **Delete Input**.

## Removing a function connection

- 1 Right-click the function connection.
- 2 Click **Delete**.

You can change the existing connection by creating a new connection. The new connection replaces the previous one.

## Duplicating an output value

You can use the output from a widget on the function modeling editor within multiple functions. For example, use a constant value as input to a Matches function and as an input into a Choose function. You can achieve this in these ways:

- Use two separate constant widgets representing the same value.
- Duplicate the output of single constant widget and connect it to the different functions.

To duplicate an output value:

- 1 Right-click the widget to duplicate the output.
- 2 Click **Duplicate Output**.

A duplicate output icon is displayed on the widget. You can now connect the output to multiple widgets.

## Removing a duplicate output

- 1 Right-click the widget to remove the output.
- 2 Click **Delete Duplicate Output**.

Where one or more function output connections are made all duplicated connections are removed.

## Adding another source

To add additional source document elements to a function:

- 1 Expand the collapsible panel to the left of the function modeling editor.  
The source document tree from visual mapping is displayed.
- 2 Drag the required document element to the function modeling editor.

## Single and List functions

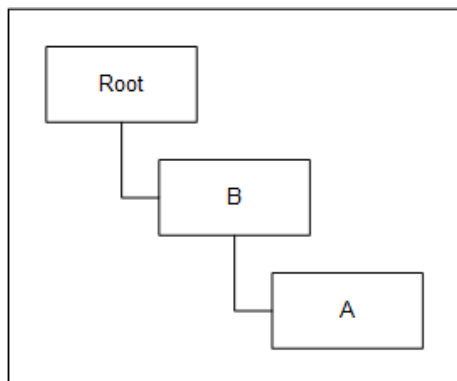
The Mapper function modeler internally handles looping constructs such as “for-each”. This makes constructing the required function simpler. Therefore, the available XSLT function tree does not contain looping construct functions.

The available XSLT functions generally expect either a single value per input parameter or a sequence of values (list) per input parameter. List functions are displayed within the Mapper function modeler available functions tree as Function name followed by an asterisk (\*). For example, Join Strings\* is a function that expects to receive a list whereas Concatenate does not.

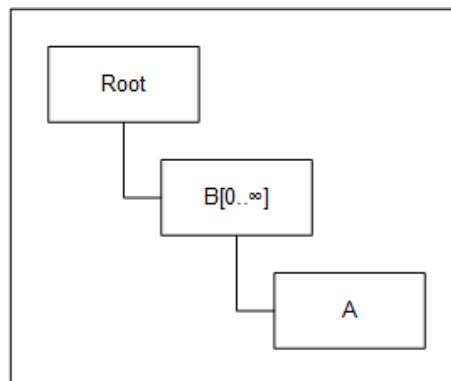
In certain modeling scenarios, the scope of the function may mean that a function that normally expects a single value requires a list as input. Additionally a function that normally outputs a single value can output a list.

Example: You can create a mapping between a source and a target entity where both source and targets are Nodes, Elements or Attributes. The initial mapping of the source and target entities is always a one-to-one mapping. That is, those entities behave as a single entity. Where multiple source entities are added within the function modeler, the additional source entities can behave as a single entity or a group entity. The relative position of the additional source (B) entity with the original source entity (A) determine its behavior as if:

- B is a parent of A, and then B behaves as a single entity.
- B is a child of parent of A and is not a list [0..∞], and then B behaves as a single entity.
- B is a child of parent of A and is a list [0..∞], and then B behaves as a group-list entity.
- B is a child of A and is not list [0..∞], then B behaves as a single entity.
- B is a child of A and is a list [0..∞], then B behaves as a group-list entity.

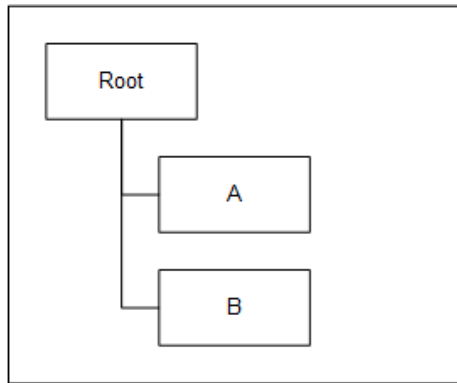


Scenario 1

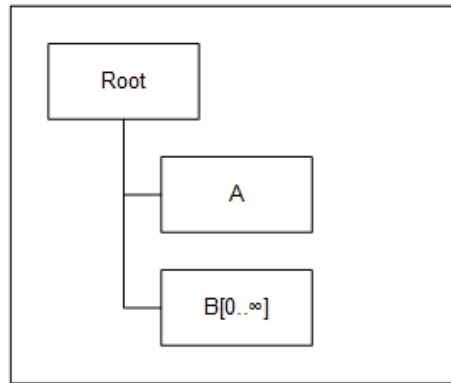


Scenario 2

A is the original source element that is mapped. Mapping additional source element B adheres to this rule: B is a parent of A; then B behaves as a single entity.



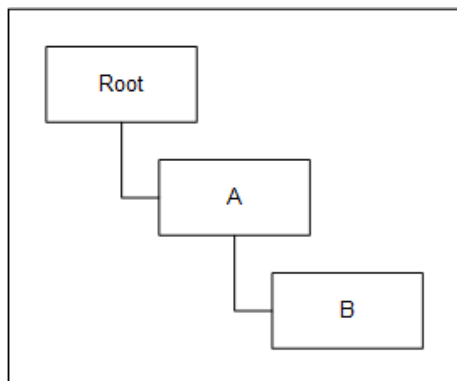
Scenario 1



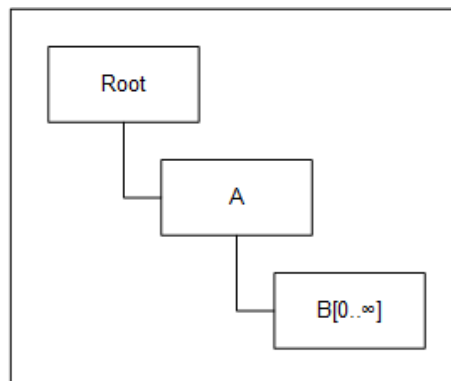
Scenario 2

A is the original source element that is mapped. Mapping additional source element B adheres to these rules:

- Scenario 1  
B and A belong to a common parent and as B is not a list, B behaves as a single entity.
- Scenario 2  
B and A belong to a common parent and as B is a list, B behaves as a group entity.



Scenario 1



Scenario 2

A is the original source element that is mapped. Mapping additional source element B adheres to these rules:

- Scenario 1  
B is a child of A and is not a list, then B behaves as a single entity.
- Scenario 2  
B is a child of A and is a list, then B behaves as a group entity.

User-Defined Functions input and out parameters can be configured to either a single value input or a list (\*) of values.

- 1 Right-click the **parameters** field.



- 2 Select Set input type as Single/List.

## Loop constructs (xsl:for-each)

When working with a list of values as the starting point of a function scope, the function modeler automatically uses the 'xsl:for-each'. This is important when defining your function if you want the scope to be different to the starting point value that is selected. When a starting point is defined as a list the function modeler uses the defined starting point for the loop. That is the 'for-each'. If the scope of the function is the element parent the alternative starting point can be selected from the widget. This changes the element used in the for-each statement.

## Removing a function

To remove a function from the graphical mapping editor:

- 1 Right-click the node in the target tree where the function is mapped.
- 2 Click **Delete Function**.  
A confirmation dialog box is displayed with the functions that are applied to the target node.
- 3 Select one or more functions that you want to delete.
- 4 Click **OK** to confirm.

## Advanced mapping

You can edit or create XSLT directly within the mapping.

Mapper is compatible with XSLT (either 1.0 or 2.0) that is created in third-party tools. There are some caveats when working with XSLT not created in mapper and some functionality is limited.

## Using advanced mapping

To use advanced mapping by importing or editing XSLT:

- 1 Click the **XSLT Editor** tab of the Mapper.  
If you have previously created a graphical mapping, the Mapper generates XSLT 2.0 based on the mapping connections and functions you have created. Lines in the generated hierarchical XSLT structure are numbered and more advanced features are available:
  - Syntax highlighting.
  - Option to validate the XSLT code and see the validation error.

- Error line highlighting and error details are available when the validation fails.
- Version comparison tool that visualizes differences in the XSLT code in two mapping version. You can copy changes from one to the other.
  - Toggle button to turn the comparison tool on or off.
  - Drop-down menu to select a version or to compare against user's custom code.
  - Side-by-side editor to compare XSLT code.
  - Right-side editor is read-only when an existing version is selected.
  - Buttons for finding previous and next difference.
  - Copy differences from the right to the left editor by clicking the arrows where a difference is detected.
- Comparison tool to evaluate the differences in the XSLT code between two mapping versions.
- Code formatting of well-formed code.
- Autocomplete, with predefined set of XSLT keywords and support for custom completions: press **Ctrl + Space**.
- Search function: press **Ctrl + F**.
  - Case-sensitive search.
  - Regular expression search.
  - Whole word search.
  - Highlight all items found.
  - Browse the page for the found elements.
- Find and Replace function: press **Ctrl+F** twice.

The same code visualization and search features are available in the **Testing** tab and in XSLT editing sections of the Function Editor. Limited options are available for non-editable fields.

## 2 Edit the XSLT.

For details, see [Editing XSLT](#) on page 155.

Alternatively, import XSLT that is created by a third party tool to manually edit the XSLT.

A header comment is inserted into the newly created or modified XSLT code while the code is in the draft state. These changes and modifications are reflected in the header comment:

- Changes that are made in the Modeler.
- Modification of these mapping properties:
  - Name
  - Description
  - Version number
  - Creation author, date and time
  - Last modification author, date and time
  - Input and output documents
- ION version number, from which the XSLT code was generated

## Importing and exporting XSLT

On the **XSLT Editor** tab of the Mapper, you can import XSLT and export XSLT. A basic format check is applied when importing XSLT. Any XSLT that is created outside of the Mapper is assumed to be

correct before it is imported. No detailed syntax checks are performed. If the imported file is not a valid XSLT file, a message is displayed and the import is not allowed. When the XSLT is imported, the **Modeler** tab is disabled. You cannot translate XSLT into a graphical mapping.

Imported XSLT must be related to documents that are registered in the document registry. If documents that are referenced in the XSLT are unavailable in the document registry you cannot perform transformations.

If necessary, to format XSLT code, click **Format**.

We recommend that you click **Validate** after importing or editing the XSLT to validate the code. If the validation fails, a message is displayed with the error line that is highlighted in the code editor.

Existing XSLT mappings from previous versions of ION can be imported as a mapping. In that case, the XSLT is regarded as an advanced mapping, because no graphical model is available. The **Modeler** tab is automatically disabled. You can move from XSLT to a graphical mapping. On the **XSLT Editor** tab, click **Switch to Graphical Mode**, that is, click the **Refresh** icon. This results in the loss of the original XSLT. A new blank graphical mapping is created on the **Modeler** tab. The graphical mapping is blank because you cannot backwards translate XSLT into a graphical mapping. This does not affect any data flows that are using the original XSLT until the data flow is re-activated.

## Editing XSLT

To edit the XSLT, set these properties:

Property	Description
XSLT	<p>The XSLT to translate the document. XSLT (Extensible Stylesheet Language Transformation) is a standard, flexible and widely supported language to transform one XML into another. Because BODs are XML documents, this language is suitable for defining a mapping.</p> <p>ION supports the 1.0 and 2.0 XSLT standards. The version is specified in the <b>version</b> attribute of the XSLT header.</p> <p>If the version is “1.0”, ION uses the Xalan XSLT processor. If the version is “2.0”, ION uses the Saxon Home Edition (HE) XSLT processor . Version “2.0” has more string and arithmetic manipulation capabilities than “1.0”. For the standard available functions, see the documentation of Saxon HE: <a href="http://www.saxonica.com/documentation/index.html#!functions/fn">http://www.saxonica.com/documentation/index.html#!functions/fn</a></p> <p><b>Note:</b> Advanced XSLT is not validated in the ION Desk. If your XSLT is incorrect, an error can occur in the ION Service when processing a BOD</p>
Strip Namespace	<p>Whether namespaces must be removed from the document. In normal cases select <b>No</b>. Select <b>Yes</b> only if your XSLT cannot deal with namespaces and namespaces are not required in the resulting document. The option to strip namespaces is available to provide compatibility with previous versions of ION (10.2 and later).</p>

## XSLT extensions

If you use an XSLT 2.0-based mapping, these extensions are provided:

- **Base64 Decoding**  
Use this extension, to decode a previously base64-encoded string inside the mapping. This extension is particularly useful when you receive files encoded using base64 standard as an input to the mapping step using XSLT. Such a file can be decoded back as a text using the XSLT extension. You can use advanced string manipulation functions to construct a complex XML output of your choice, including hierarchies. The mapper does not validate if the input is an already base64-encoded string and decodes it using base64 decryption. If you use this extension, ensure that the input string is already base64-encoded.
- **Encode Input String**  
Use this extension to encrypt any string to a base64 standard inside a mapping. You can encrypt any field in your BOD as a base64 string.
- **Raise Exceptions**  
Use this extension, to perform business or metadata validations in your XSLT code and raise exceptions. When this extension is called, the mapping step throws a confirm BOD with the specified reason, and the execution of the data flow is terminated.

This section provides example use cases and sample XSLT code on how to use these extensions.

## Namespace for custom extensions

The custom extensions are defined in the "http://www.infor.com/ION/XSL/extensions" namespace. You must include them in the header of XSLT code where the extensions are used. For example:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:extensions="http://www.infor.com/ION/XSL/extensions" version="2.0">
```

Normally the output of an XSLT where extensions are used also includes this namespace. For example, a custom BOD called "SyncMyItem" produced from an XSLT with extension contains this code:

```
<SyncCustomItem xmlns:extensions=http://www.infor.com/ION/XSL/extensions>
```

To suppress this namespace in the output XML, include the "exclude-result-prefixes" attribute in the header of the XSLT. For example:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:extensions="http://www.infor.com/ION/XSL/extensions" exclude-result-
prefixes="extensions" version="2.0">
```

## Example for base64 decoding extension

Any file transported by the file connector in binary format is encoded using base64 standard and copied inside the `<RawData>` tag in the noun section of the DataArea. The Base64 decoding extension is useful to decode such a file using an XSLT mapping. If the original file is a text file, you can use advanced string manipulation functions on the decoded text and construct a complex XML output of your choice including hierarchies.

The following example illustrates how you can use a plain text file to construct an XML with hierarchical structure.

**Note:** This example is included only as an illustration. It is not intended as an optimal or advised approach.

## Example scenario

Order information is published by a third party application as a text file in a header-line detail format. You must transform this information to a `Sync.PurchaseOrder` XML document and send it to an Infor application. To achieve this:

- 1 Create these file format templates in ION:
  - A binary format with the custom noun called 'MyRawData'
  - A Full BOD format with the standard noun 'PurchaseOrder'
- 2 Read the text file as 'Sync.MyRawData' BOD using the ION file connector.
- 3 Send it to an XSLT mapper step.
- 4 Use the decode extension of XSLT mapper to convert the base64 file data to a string.
- 5 Use the advanced string manipulation and logical functions of XSLT 2.0 to construct the `Sync.PurchaseOrder` XML document.

The output of the mapper step is a `Sync.PurchaseOrder` XML that you can send to the Infor application.

## Original file sample

This is a sample of an order information file:

```
po123,ACME_Electronics,EUR,5000^501?Computers?2^502?Printers?3|po234,ACME
Furnitures,USD,2000^601?Tables?1^602?Chairs?4
```

This example file contains two records in a header-line format with these delimiters as separators:

- | is the row separator.
- , is the header field separator.
- ^ is the order line separator.
- ? is the order line's field separator.

## Sample after read action by file connector

In the first step, this file is read by the file connector as a custom BOD called 'Sync.MyRawData'. This is a sample of the resulting BOD in ION:

```
<?xml version="1.0" encoding="UTF-8"?>
<SyncMyRawData releaseID="9.2">
  <ApplicationArea>
    <Sender>
      <LogicalID>infor.file.read</LogicalID>
      <ComponentID>External</ComponentID>
      <ConfirmationCode>OnError</ConfirmationCode>
    </Sender>
    <CreationDateTime>2014-12-09T11:00:28.521Z</CreationDateTime>
    <BODID>infor.file.read:1418122828521:10a4be73-3377-4555-a017-52e911e4b88b</BODID>
  </ApplicationArea>
  <DataArea>
    <Sync>
      <TenantID>INFOR</TenantID>
      <AccountingEntityID/>
      <LocationID/>
      <ActionCriteria>
        <ActionExpression actionCode="Replace"/>
      </ActionCriteria>
    </Sync>
    <MyRawData FileName="exampleFile1" FileExtension="txt">
      <Id>RD-677911891-1418122828521</Id>
      <RawData>cG8xMjMsQUNNRSBFbGVjdHJvbmljcyxFVVI sNTAwMF41MDE/Q29tcHV0ZXJzPz
JeNTAyP1ByaW50ZXJzPzN8cG8yMzQsQUNNRSBGdXJuaXR1cmVzLFVTRCwyMDAwXjYwMT9UY
WJsZXM/MV42MDI/Q2hhaXJzPzQ=</RawData>
    </MyRawData>
  </DataArea>
</SyncMyRawData>
```

## XSLT sample

Use this sample to transform the Sync.MyRawData BOD XML to a Sync.PurchaseOrder BOD XML Document.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:extensions="http://www.in
for.com/ION/XSL/extensions" exclude-result-prefixes="extensions" ver
sion="2.0">
  <xsl:output method="xml" indent="yes"/>
  <xsl:template match="SyncMyRawData">
    <SyncPurchaseOrder>
      <xsl:apply-templates select="ApplicationArea"/>
      <xsl:apply-templates select="DataArea"/>
    </SyncPurchaseOrder>
  </xsl:template>

  <xsl:template match="ApplicationArea">
    <xsl:element name="ApplicationArea">
```

```

    <xsl:copy-of select="*" />
  </xsl:element>
</xsl:template>

<xsl:template match="DataArea">
  <xsl:element name="DataArea">
    <xsl:apply-templates select="Sync" />
    <xsl:apply-templates select="MyRawData/RawData" />
  </xsl:element>
</xsl:template>

<xsl:template match="Sync">
  <xsl:element name="Sync">
    <xsl:copy-of select="*" />
  </xsl:element>
</xsl:template>

<xsl:template match="MyRawData/RawData">
  <xsl:variable name="b64" select="extensions:decode
String(string(.))" />
  <xsl:for-each select="tokenize($b64, '\\|')">
    <xsl:variable name="Header" select="." />
    <PurchaseOrder>
      <PurchaseOrderHeader>
        <DocumentID>
          <xsl:value-of select="tokenize($Header, ',') [position()
= 1]" />
        </DocumentID>
        <SupplierParty>
          <Name>
            <xsl:value-of select="tokenize($Header, ',') [position()
= 2]" />
          </Name>
        </SupplierParty>
        <BaseCurrencyAmount>
          <Amount CurrencyID="{tokenize($Header, ',') [position() =
3]}">
            <!-- Business Validation Checks -->
            <xsl:variable name="Amount" select="tokenize($Head
er, ',') [position() = 4]" />
            <xsl:variable name="Amount" select="substring-be
fore($Amount, '^)" />
            <xsl:value-of select="$Amount" />
          </Amount>
        </BaseCurrencyAmount>
      </PurchaseOrderHeader>
      <xsl:for-each select="tokenize($Header, '\\^'">
        <xsl:variable name="Line" select="." />
        <xsl:if test="position() != 1">
          <PurchaseOrderLine>
            <ItemID>
              <xsl:value-of select="tokenize($Line, '\\?') [position() =
1]" />
            </ItemID>
            <Description>
              <xsl:value-of select="tokenize($Line, '\\?') [position() =

```

```

2]"/>
        </Description>
        <Quantity>
          <xsl:value-of select="tokenize($Line,'\?')[position() =
3]"/>
        </Quantity>
      </PurchaseOrderLine>
    </xsl:if>
  </xsl:for-each>
</PurchaseOrder>
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

### Final output sample

This transformation results in this final output:

```

<?xml version="1.0" encoding="utf-8"?>
<SyncPurchaseOrder xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <ApplicationArea>
    <Sender>
      <LogicalID>infor.file.read</LogicalID>
      <ComponentID>External</ComponentID>
      <ConfirmationCode>OnError</ConfirmationCode>
    </Sender>
    <CreationDateTime>2014-12-09T11:00:28.521Z</CreationDateTime>
    <BODID>infor.file.read:1418122828521:10a4be73-3377-4555-a017-
52e911e4b88b</BODID>
  </ApplicationArea>
  <DataArea>
    <Sync>
      <TenantID>INFOR</TenantID>
      <AccountingEntityID />
      <LocationID />
      <ActionCriteria>
        <ActionExpression actionCode="Replace" />
      </ActionCriteria>
    </Sync>
    <PurchaseOrder>
      <PurchaseOrderHeader>
        <DocumentID>po123</DocumentID>
        <SupplierParty>
          <Name>ACME_Electronics</Name>
        </SupplierParty>
        <BaseCurrencyAmount>
          <Amount CurrencyID="EUR">5000</Amount>
        </BaseCurrencyAmount>
      </PurchaseOrderHeader>
      <PurchaseOrderLine>
        <ItemID>501</ItemID>
        <Description>Computers</Description>
        <Quantity>2</Quantity>
      </PurchaseOrderLine>
      <PurchaseOrderLine>

```



```

    <ItemID>502</ItemID>
    <Description>Printers</Description>
    <Quantity>3</Quantity>
  </PurchaseOrderLine>
</PurchaseOrder>
<PurchaseOrder>
  <PurchaseOrderHeader>
    <DocumentID>po234</DocumentID>
    <SupplierParty>
      <Name>ACME Furnitures</Name>
    </SupplierParty>
    <BaseCurrencyAmount>
      <Amount CurrencyID="USD">2000</Amount>
    </BaseCurrencyAmount>
  </PurchaseOrderHeader>
  <PurchaseOrderLine>
    <ItemID>601</ItemID>
    <Description>Tables</Description>
    <Quantity>1</Quantity>
  </PurchaseOrderLine>
  <PurchaseOrderLine>
    <ItemID>602</ItemID>
    <Description>Chairs</Description>
    <Quantity>4</Quantity>
  </PurchaseOrderLine>
</PurchaseOrder>
</DataArea>
</SyncPurchaseOrder>

```

## Example for throw exception extension

In the previous example of decode base64 string extension, you can perform additional business validations. For example, you can verify whether the value of the 'Amount' element in the purchase order is an integer. If not, you can throw an exception using the 'throwException' extension. If this extension is called, the mapping step throws a confirm BOD with the specified reason and the execution of the document flow is terminated

To invoke this extension, use this XSLT sample:

```

<xsl:if test="not($Amount castable as xs:integer)">
  <xsl:comment select="extensions:throwException('Extended Amount value
  is not an integer')"/>
</xsl:if>

```

## Example for base64 encoding extension

This extension is useful if you want to encrypt any field in your BOD as a base64 string. For example, use the following XSLT sample to assign any base64-encoded string to an element called 'EncodedData'.

```
<EncodedData>  
  <xsl:value-of select="extensions:encodeString('An Encoded String')"/>  
</EncodedData>
```

## Switching back to graphical mapping

If you previously created a graphical model, you can revert to the XSLT that represents that graphical model. In that case, the manually added XSLT is overwritten. You cannot 'reverse engineer' the manually written or imported XSLT back into a visual mapping.

To revert to graphical mapping:

- 1 Create a backup copy of the XSLT, to avoid loss of the original XSLT.  
Either use the Export option, or duplicate the mapping from the main menu to create a new working copy.
- 2 Click the **XSLT Editor** tab.
- 3 Click **Switch to Graphical Mode**.
- 4 View or edit the graphical mapping.

## Sensitive data mapping security policies

When you model a sensitive data mapping, you can define these security policies on selected nodes:

- Data masking - replaces the data value of an element or attribute in the output document with a string. If you use data masking, you must specify at least one character for the masking value.
- Data removal - removes the value of an element or attribute in the output BOD. This produces an empty node with an empty string.
- Node removal - removes the node and its children from the output document.

If a security policy is applied on a parent node, it removes the child elements of that parent node.

The masking table provides information about added nodes and the security policies applied on those nodes. For each added node, this information is shown:

- Status.
- Node name.
- Node path.
- Node data type.
- Security policy.

- Masking value of the data masking security policy.

You can filter the masking table using keywords in the masking table search field. For example, you can filter all the nodes that have a certain keyword as their masking value.

When the node is focused in the document tree, it is highlighted and expanded to show its children. This allows you to continue adding the parent, children or sibling nodes of the node that is in focus.

## Creating a security policy

To create a security policy for a node:

- 1 Select a node in the document tree.
- 2 Right-click and select **Add to Masking Table**. The node is added as a row to the masking table.
- 3 In the **Security Policy** column, select the security policy from the drop-down box.
- 4 Click **OK**.

## Changing applied security policy

To change already added security policies on a node:

- 1 Search for the row in the masking table that contains the node to change the security policy.
- 2 In the **Security Policy** column, select the drop-down box and select the new policy to apply.

## Creating and updating security policy on multiple nodes

To update multiple nodes with a single security policy:

- 1 Add nodes from the document tree to the masking table.
- 2 Select the nodes in the masking table to create or update a security policy.  
When a node is selected, the masking table toolbar is displayed.
- 3 Click **Edit** in the masking table toolbar.
- 4 Select the security policy that must be applied to multiple nodes.
- 5 Click **OK** to confirm.  
The security policy is applied to all the selected nodes and any previous security policy is overwritten.

---

## Applying the Data Masking security policy

Use the Data Masking security policy to replace sensitive data by a string of characters, instead of completely removing it from the output.

**Note:** If you mask parent nodes that contain child elements, those child elements are removed from the output.

- 1 Select a node in the document tree. Then, right-click and select **Add to Masking Table**.
- 2 Select the drop-down box in the **Security Policy** column.  
The available security policies are displayed.
- 3 Select **Data Masking** from the drop-down box.
- 4 In the **Masking Value** column, specify the data masking string.  
For a valid data masking value, type at least one character.  
The applied masking values can be searched in the masking table.

## Applying the Data Removal security policy

Use the Data Removal policy to remove the value of an element or attribute from the output document. The element is not removed from the output, but only its value.

- 1 Select a node in the document tree. Then, right-click and select **Add to Masking Table**.
- 2 Select the drop-down box in the **Security Policy** column  
The available security policies are displayed.
- 3 Select the **Data Removal** policy from the list.

**Note:** If you apply the Data Removal policy on parent nodes that contain child elements, those child elements are removed from the output.

The masking value column is disabled for the Data Removal security policy.

## Applying the Node Removal security policy

Use the Node Removal policy to remove a node and its children from the output.







- 1 Select a node in the document tree. Then, right-click and select **Add to Masking Table**.
- 2 Select the drop-down box in the **Security Policy** column.  
The available security policies are displayed.
- 3 Select **Node Removal** policy from the list.

**Note:** The masking value column is disabled for the Data Removal security policy.

## Security policy status and feedback

For each added node to the masking table you can review the status of the security policy. Visual feedback is provided through icons and tooltips with detailed messages.

This table shows the icons and their description:

icon	Status	Tooltip message	Description
	Warning	Security policy is applied on a node that is required.	A required node is added to the masking table and a security policy is applied. You can approve this mapping.
	Warning	This node is not included in the output. Its parent node is set for node removal.	Displayed on child nodes, if a parent node is added to the masking table, and any of its added children are added to the grid. You can approve this mapping.
	Warning	This node is not included in the output. A security policy is applied to one of its higher-level parent nodes.	This warning is shown if node attributes were added to the masking table. Their grandparents are added to the masking table with a security policy applied.
	Warning	This node is not included in the output. Its parent node is set for node removal.	Displayed on an attribute that was added to the masking table and its parent node has a security policy "Node Removal" applied.
	Error	No security policy is applied. This mapping cannot be approved.	A node is added to the masking table, but no security policy is applied on the node. This mapping cannot be approved. All added nodes to the masking table must have a security policy applied.
	Confirmed	Policy is applied.	A node is added to the masking table and a security policy is applied. This is a valid status and you can approve the mapping.

## Removing a security policy for a node

- 1 Select one or more rows in the masking table.  
The masking table toolbar is displayed.
- 2 Click **Delete**.

The selected nodes and the security policies that are applied on them are removed from the masking table.

## Testing a mapping

Complete these steps:

- 1 Go to the mapping editor.
- 2 Click the **Testing** tab.
- 3 Select a valid input XML document.
- 4 Click **Preview**.

The XML is submitted to the runtime. After processing through the XSLT the output of the document is displayed.

If an error is returned or the target document does not contain the information expected, make appropriate changes to your mapping. Click **Preview** again.

## Comparison tool

To visualize the differences between the test input and output documents, you can use the comparison tool.

In the comparison tool you can use these functionalities:














- Toggle to switch the comparison tool on and off.
- Match and highlight the differences in the documents.
- Scroll through the differences by clicking the up and down icons.
- Apply matched differences from the test output document into the input document. Click the left arrow icon next to the chosen difference.

To ensure proper matching of differences between the test input and output documents:

- 1 Click **Preview** to create an output document.
- 2 Click **Compare** to switch on the comparison tool.
- 3 Click **Format** above both input and output fields to apply the same code formatting.


## Icons and visual feedback

This table shows the main modeling icons used within a graphical mapping:

Icon	Functional Area	Description
	Graphical Mapping	XSD node
		XSD element
		XSD attribute
		A node, element or attribute is completely matched
		A node or element matched by a function on a parent node. The applied function returns an 'anyType' output. This means that the function is providing all the data for the children nodes/elements/attributes, causing the badges on all of the children to appear as a green circle.
		Unmatched optional node, element or attribute
		Warning – a non-mandatory node, element or attribute is partially matched Warning – an element or attribute is overloaded: it has two or more matches from the source document Warning – a node is not explicitly matched, but a link is inferred from the connection of one or more child elements
		Error – a mandatory node, element or attribute is not completely matched
		Function exists for the node or element
		Multiple functions exist for node or element
	Function Modeling	Function output duplicate
	Graphical Mapping	Default value exists for the node or element
	Graphical Mapping	Comment was attached to specific mapping relation or container


The term 'pseudo-matched' means that the match is done based on inference. The node is not explicitly matched, but a connection is derived from the connection of one or more child elements.





This table explains the detailed context of the connection icons displayed in a graphical mapping.

Mandatory	Matched	Has mandatory children	Mandatory children matched	Has non-mandatory children	Non-mandatory children matched	Result	Icon
Yes	No	No	-	No	-	Mandatory unmatched	

Mandatory	Matched	Has mandatory children	Mandatory children matched	Has non-mandatory children	Non-mandatory children matched	Result	Icon
Yes	No	No	-	Yes	No	Mandatory unmatched	!
Yes	No	No	-	Yes	Yes	Mandatory unmatched	!
Yes	No	Yes	No	No	-	Mandatory unmatched	!
Yes	No	Yes	No	Yes	No	Mandatory unmatched	!
Yes	No	Yes	No	Yes	Yes	Pseudo-matched	!
Yes	No	Yes	Yes	No	-	Pseudo-matched	!
Yes	No	Yes	Yes	Yes	No	Pseudo-matched	!
Yes	No	Yes	Yes	Yes	Yes	Pseudo-matched	!
Yes	Yes	No	-	No	No	Matched	✓
Yes	Yes	No	-	Yes	No	Mandatory unmatched	!
Yes	Yes	No	-	Yes	Yes	Matched	✓
Yes	Yes	Yes	No	No	-	Mandatory partially matched	!
Yes	Yes	Yes	No	Yes	No	Mandatory partially matched	!
Yes	Yes	Yes	No	Yes	Yes	Mandatory partially matched	!
Yes	Yes	Yes	Yes	No	-	Matched	✓



Mandatory	Matched	Has mandatory children	Mandatory children matched	Has non-mandatory children	Non-mandatory children matched	Result	Icon
Yes	Yes	Yes	Yes	Yes	No	Partially matched	
Yes	Yes	Yes	Yes	Yes	Yes	Matched	
No	No	No	-	No	-	Un-matched	
No	No	No	-	Yes	No	Un-matched	
No	No	No	-	Yes	Yes	Pseudo-matched	
No	No	Yes	No	No	-	Un-matched	
No	No	Yes	No	Yes	No	Un-matched	
No	No	Yes	No	Yes	Yes	Mandatory partially matched	
No	No	Yes	Yes	No	-	Pseudo-matched	
No	No	Yes	Yes	Yes	No	Pseudo-matched	
No	No	Yes	Yes	Yes	Yes	Pseudo-matched	
No	Yes	No	-	No	-	Matched	
No	Yes	No	-	Yes	No	Partially matched	
No	Yes	No	-	Yes	Yes	Matched	
No	Yes	Yes	No	No	-	Mandatory partially matched	
No	Yes	Yes	No	Yes	No	Mandatory partially matched	

Mandatory	Matched	Has mandatory children	Mandatory children matched	Has non-mandatory children	Non-mandatory children matched	Result	Icon
No	Yes	Yes	No	Yes	Yes	Mandatory partially matched	
No	Yes	Yes	Yes	No	-	Matched	
No	Yes	Yes	Yes	Yes	No	Partially matched	
No	Yes	Yes	Yes	Yes	Yes	Matched	

## Versioning

A mapping can have multiple versions. The Mapper and ION Desk can track these versions internally to detect which version of a mapping is used by a data flow model. Multiple versions of a mapping in the approved state may be in use in several different data flow models at any point in time.

A mapping can be an approved version, such as V1 or V2, or a draft in the editable state. Activation of a mapping is done outside of the Mapper tool.

The user validates the latest available draft by invoking the approve method in the Mapper. An approved mapping can be used in an active data flow.

If the latest mapping version is approved, it is locked and the major version number gets incremented by +1 from the previous version. For example: Version 1 -> Version 2.

If the latest version is currently a full version, which is Active-Approved, the version information remains unchanged. Active-Approved means this version has not been edited since its last activation and no current draft exists.

## Version administration

The **Version** widget provides a list of draft and approved versions. For each of the versions, the latest date & time and name of the author who changed the version are displayed.

These instructions explain individual steps in version management:

- To create the first draft, create and save a new mapping.
- To edit the current draft, open and modify the mapping. Then save the changes.
  - When a draft version has unsaved changes, an indicator, a yellow triangle, is displayed on the draft version in the version list of the mapping.

- To approve the current draft, click the check mark in its line in the **Versions** widget. You can also approve single or multiple mappings from the Mapper overview page.
  - If the draft has unsaved changes, you are prompted to save the changes before proceeding with the approval of the draft.
- To edit an approved version, click the pencil icon. A new draft is created. The version number that is assigned to the draft is incremented from the latest approved version number, existing or already deleted. Each version number is unique.
- To delete an approved version, click the bin icon. You can delete only an inactive approved version. To verify whether the version is inactive, check the active/inactive identifier above the widget. The system always requires that at least one draft or one approved version exists.
- For activation of an approved version of a mapping, the respective mapping must be saved properly.

## Approving draft versions

Draft versions of mapping(s) can be approved from the overview page and from inside a mapping.

To approve multiple mappings from the overview page:

- 1 Select more than one mapping that must be approved.
- 2 Click the check mark icon for approval in the main toolbar. This approves the selected mappings if they are valid. Skip the approval if they are already approved or return an error message when a mapping cannot be approved.

To approve a single mapping from the overview page:

- 1 Move the pointer over the mapping tile and click the check mark icon in the tile toolbar.
- 2 From the overview page, mappings that cannot be approved display an error message.

To approve a mapping from the mapping view, click the check mark icon in the widget panel.

Graphical model mappings are not approved if they contain any errors shown in the expanding notification panel.

Code only mappings are not approved if they do not pass the XSLT validation check.

When a mapping cannot be approved, an error message is displayed and the code editor highlights the invalid line in the code.

Sensitive data mappings cannot be approved if the nodes that were added to the masking table have no security policy applied on them.

Successfully approved mapping(s) lock the mapping name and source/target documents, meaning they cannot be changed.

## Version and approval status in the Mappings overview page

In the **Mappings** overview page, you can inspect the approval status and the latest approved version number on each of the mapping tiles. Additionally, you can filter the mappings by their approval status.

In the **Mappings** overview page, on each mapping this information is displayed:

- Approval status of a mapping. Displayed as a status badge on the mapping tile.  
In the Grid view, the approval status is displayed as a read-only check box in the respective columns.
- The version number of the latest approved version in the mapping.

Additionally, you can use the filter to show only the mappings that have an approval status that matches the selection. To filter the list of mappings by their approval status, select one or more of these options:

- Draft - mapping with only one version that is not approved
- Approved - mapping with only approved versions and no draft
- Approved with a Draft - mapping with approved versions and a draft

---

## Chapter 6: ION Scripting

With ION Scripting, integration developers, administrators, and analysts can write custom Python code as part of their business process integrations.

**Note:** ION Scripting is not available in AWS GovCloud.

These scripts can be used as a data transformation activity within an ION Document Flow or Data Lake Flow. ION Scripting can be used to facilitate and solve common use cases such as:

- Data object format conversion
- Data mapping
- Complex calculations & transformations
- Simplified data manipulation orchestration

To use ION Scripting, expand the ION Desk menu, select **Scripting**, and click either **Scripts** or **Libraries**.

### ION Scripting concepts

ION Scripting is separated in two primary components of data transformation management; Scripts and Libraries.

Scripts provide a functional management page and browser-based development IDE for writing or importing Python scripts. These scripts can then be referenced and managed as an activity within ION's Data Flow modelers.

For documentation on how to create and manage an ION script, see [Script modeling](#) on page 182.

Libraries provide a management component for importing and storing custom and open source Python packages that can be referenced within an ION script. Libraries can be useful in accelerating development to leverage functional packages that may complete or automate an otherwise complex business requirement.

For documentation on how to manage libraries, see [Libraries](#) on page 180.

Scripts enable complex data transformation and calculation scenarios for business process integrations. ION Scripting does impose limitations on data processing times and payload processing sizes.

For more details on using scripts and the limitations, see [Using scripting](#) on page 189.

ION Scripting provides a robust suite of tools and developer-oriented features to simplify the process of writing and coding scripts. Yet it may not provide developer features associated with local IDE clients and applications.

For complex coding use cases or advanced developers, developing scripts within an IDE and importing them into ION Scripting can be a preferred development workflow.

Note that script execution, performance, and behavior can differ between a local IDE client and ION Scripting. Sufficient testing must be done to ensure consistent development expectations when deploying a script within a Data Flow. When you are developing code outside Scripting, ensure the code and libraries you work with are supported.

See [Supported languages](#) on page 174.

## Supported languages

ION Scripting supports the use of libraries and writing script code in Python 3.7.

We recommend that libraries and scripts are written and designed to work with the specified Python version. Scripts that are developed in earlier versions prior to Python 3, such as Python 2.7, are not supported.

## Supported libraries

A supported library distribution package format is Python Wheels (Wheel Binary Package Format).

Python Wheels is a common open-source format that is a standard for Python package distribution.

These are the support Python Wheels compatibility tags in ION Scripting:

- Python version tag: py2.py3, py3, py36, py37, cp37
- Platform tag: manylinux1\_x86\_64, any.
- ABI tag: cp37m

This table lists examples of valid and invalid file names:

File name	Supported	Reason
pandas-0.25.1-cp37-cp37m-manylinux1_x86_64.whl	Yes	
numpy-1.17.1-cp36-cp36m-manylinux1_x86_64.whl	No	Python version cp36 and ABI tag cp36m are not supported
numpy-1.17.1-cp36-cp36m-macosx_10_9_x86_64.whl	No	Platform tag macOS is not supported.
lxml-4.4.1-cp37-cp37m-win32.whl	No	Platform tag win32 is not supported.

File name	Supported	Reason
lxml-4.4.1-cp27-cp27m-manylinux1_x86_64.whl	No	Version tag cp27 and ABI tag cp27m are not supported

For more details on scripting libraries, see [Libraries](#) on page 180.

## Scripts overview page

On the Scripts overview page you can:

- Add new script.
- Delete script(s).
- Duplicate script.
- Export and Import script(s).
- Approve script(s).
- Check the Usage of a script.
- Search by keywords.
- Filter by approval status.
- Change the view between the Tiles view (default) and the Grid view.

## Creating scripts

To create a new script, click the **Add+** tile labeled in the Tiles view or the **+** icon in the data grid in the Grid view.

For more information, see Script Modeling.

## Deleting scripts

- 1 Hover over a tile to view the tile's action toolbar.
- 2 Click **Delete**.

You can use **Select** to select multiple scripts and delete them. In the Grid view, select one or multiple rows and click **Delete**.

**Note:** Script management behaves slightly different than the other Infor ION content. Scripts can be deleted regardless of whether that script is actively used in a Data Flow.

---

## Duplicating scripts

- 1 Hover over a tile or select one row.
- 2 Click **Duplicate**.

The latest version of the selected script and its contents is copied to a new script.

When duplicating a script, any sample files or content pasted into the **Testing** tab is not migrated to the new script.

## Exporting scripts

You can export one or more scripts and save them as a JSON file export.

- 1 Hover over a tile to view the tile's action toolbar.
- 2 Click **Export**.

You can use **Select** to select multiple scripts and export them. In the Grid view, select one or multiple rows and click **Export**.

During export, any sample files or input content used during testing is not included in the file export process.

## Importing scripts

- 1 Click **Import**.
- 2 Select a local scripts export file (JSON file format).

The new scripts that did not exist in ION Scripting are imported and displayed in the Overview page as the newest ones.

If an already existing script that must be imported is detected in ION Scripting, you are asked to select a preferred import behavior. A prompt presents any duplicate scripts including the name of the script and its description and options to perform an import action.

You can select one of these options:

- Append as Draft. Select the imported duplicate script(s) in the grid that must be appended as draft to the corresponding existing ones.

**Note:** Draft versions on the existing script(s) are overwritten. If there is no draft version, a new one is created.

- Skip importing the non-selected scripts in the grid.
- Rename the non-selected duplicate scripts after importing.

If you do not select any scripts in the data grid, you can choose to skip or rename all imported duplicates.

- 3 Click **OK** to confirm your choice or click **CANCEL** to abort the import.



---

## Script details

Script details are displayed on the tiles in the Tiles view, or in the data grid in the Grid view.

For each script, these details are displayed:

- Script name
- Script description
- Script version status
- Script latest approved version number
- Last updated date
- Last updater's username

## Approving scripts

- 1 Hover over a tile to view the tile's action toolbar.
- 2 Click **Approve**.

You can use **Select** to select multiple scripts and approve them. In the Grid view, select one or multiple rows and click **Approve**.

After the approval action is completed, a status dialog box is displayed. For each script that was selected the status is displayed. The status' can indicate this information:

- Successfully approved the draft version in a script.
- No draft versions were found to be approved for a script.
- Error occurred while approving the draft version, because the script is invalid.

In this scenario, open the script to investigate and solve the issue.

**Note:** For versioning and approving scripts, see Script Modeling.

## Script usage

You can identify and understand how ION scripts are deployed and from which data flows those scripts are being used as an activity step.

- 1 Hover over a tile or select one row.
- 2 Click **Usage**.

Usage indicates which document flow(s) or Data Lake flow(s) are currently using the script.

- 3 Click on a flow name and drill down to the modeling page of the flow.

## Searching and filtering scripts

- 1 Click **Search**.

- 2 Specify a search string.  
The search string is matched against the script name, script description and last updater's username.
- 3 Click **Filter** to filter scripts by script version status and select one of these types:
  - Draft: Scripts that have only a draft version and no approved versions.
  - Approved with a Draft: Scripts containing a draft version and approved versions.
  - Approved: Scripts that have all versions approved.

## Libraries overview page

On the Libraries Overview page you can:

- Import a library.
- Delete a library or libraries.
- Export a library.
- Check the Usage of a library.
- Search by keywords.
- Change the view between the Tiles view (default) and the Grid view.

## Importing libraries

The supported file format extension is Python Wheels - `.whl`

You can manage and use different versions of the same library in scripts. You cannot import a library with an existing name and version in the system.

On import these details are retrieved from the file:

- Library Name
- Description
- License
- Author
- External Dependencies to other libraries
- Contents of the library

For more details on supported libraries and library versioning, see [Libraries](#) on page 180.

To import a library:

- 1 Click **Import**.
- 2 Select a local file to upload.

If you import an existing library with a different release version, a confirmation dialog box is displayed. You are informed that the library version was added to the existing one.

---

## Deleting libraries

- 1 Hover over a tile to view the tile's action toolbar.
- 2 Click **Delete**.

You can use **Select** to select multiple libraries and delete them. In the Grid view, select one or multiple rows and click **Delete**.

**Note:** You cannot delete a library that is used in an approved script version. To delete such a library, first remove the library dependency from the script.

## Exporting libraries

You can export one or more scripts and save them as a JSON file export.

- 1 Hover over a library tile to view the tile's action toolbar.
- 2 Click **Export**.

The export file is a Python Wheel package that was used to import the library.

Libraries must be exported individually. You cannot export multiple libraries.

## Library details

For each library, these details are displayed on the tiles in the Tiles view, or in the data grid in the Grid view:

- Library name
- Library description
- Last updated date
- Last updated by

## Library usage

You can identify and understand how ION libraries are deployed and from which script those libraries are being used.

- 1 Hover over a library tile.
- 2 Click **Usage**.

The usage dialog box displays these details for each library version:

- Name of script using the library version
- Script description
- Script version number
- Script version status

**Note:** Missing libraries versions are also shown in case a script is using a version that was deleted or never uploaded. These missing Library versions are indicated with an error status in the **Library Usage** dialog box.

Multiple versions of a library cannot be used in a single script version.

You can check the Library usage from within the **Library details** page, by clicking **Usage**.

## Searching libraries

1 Click **Search**.

2 Specify a search string.

The search string is matched against the library name, script description and last updater's username.

## Libraries

From the libraries overview page, click one of the library names to open the details page.

The library details page consists of these elements:

- Library's left panel with the name, description and contents of the library.
- Versioning widget for managing multiple imported library versions.
- Information tab with general information about the library.
- Read-only code editor to browse the contents of the library.

The library details are parsed from the library package itself. Attributes and details describing the library may be unavailable or otherwise not listed if the package developer did not provide these details.

In the library details page, you have these options:

- Change the description of the library.
- Export the library to a `.whl` file.
- Check the library usage.

Libraries are considered read-only with the exception of the description field. ION administrators may append a custom description to help describe library functionality and usage.

Note that the version of the library exported from ION is the version currently selected and viewed within the ION Scripting UI. To export a different version of the library, select a previous version to view before exporting.

## Information tab

The **Information** tab shows library details that were stored as part of the distribution package.

Python Wheels is a common standard that requires from package authors to provide general details about the distribution.

The **Information** tab shows these library details:

- Package distribution name
- Version
- License
- Home page URL
- Author
- External library dependencies

The dependencies indicate what other libraries you must import in the system for this library to work.

If an external library dependency is missing for a library, or an incorrect version is available. You are notified with a Warning icon on the affected external library dependency in the **Information** tab. The same warning indicator is shown in Scripts details, if you add a library to a script that has missing or incorrect external library dependencies.

## Browsing the library content

You can explore the contents of a library to understand how that library should be used in your code.

To view files of the library:

- 1 Expand the left panel if it is collapsed.
- 2 Click the library name in the contents tree.
- 3 Click the file to view.

The file is opened in a read-only code editor in a new tab. You can have multiple files opened in different tabs. Click the X icon to close the tab. Additionally, right-click a tab and select **Close All** to close all file tabs. The Information tab cannot be closed.

The read-only code editor supports syntax highlighting and searching.

## Searching the code editor

To search in the code editor:

- 1 Click in the editor.
- 2 Press CTRL + F on your keyboard.  
The search bar is displayed.
- 3 Specify the keyword to search for.  
The matched keyword is focused and highlighted.

- 4 Use the find next and find previous arrows to find the next match.

## Library versioning

Multiple versions of the same library can be maintained and used in scripts.

A script can use only one version of a library when creating a dependency. The choice of the version that is used can be changed in the script.

The library Versions widget displays all the available versions for the open library. The current open library version is highlighted and in focus. When you open a library from the Overview page, the latest version of the library is displayed. To explore previous versions of a library, click different versions within the widget.

Details displayed for a library version in the library Versions widget:

- Version number
- Import date and time
- Name of the user that imported the version

## Finding and creating libraries

Depending on your organization, libraries can be provided by different sources or distributed in different formats

ION Scripting supports libraries in Python Wheels format - a common open-source format that is a standard for Python package distribution.

The Python Package Index (PyPI) online repository offers free access to a large number of Python libraries. You can use the Python Wheels format in ION Scripting.

Libraries that you have developed must be built as distribution packages in a Python Wheel format.

Python packages that are in a `tar.gz` or `.zip` file format can contain C extensions. These extensions require to be compiled in a Python environment that is matching the ION Scripting environment. Libraries that are in `.tar.gz` or `.zip` file format are not supported and must be built as Python Wheels.

When building Python Wheel packages, ensure the build environment is:

- Python 3.7
- Linux x86\_64

## Script modeling

Creating a new script from the **Overview** page or opening an existing script, takes you to the Script details page where you define the script.

---

The Script details page consists of these elements:

- Script's left panel where you provide the script name and description.
- Versioning widget for managing the script versions.
- Code editor for writing the script code and for viewing the files from the libraries.
- Library dependencies widget to add libraries to the script.
- Input Variable and Output Variable names definition fields.
- Performance Metrics showing details of a test performance.
- Documentation tab for writing rich-text documentation for the script.

We recommend this workflow pattern for modeling scripts:

- Define the script name.
- Define the Input Variable and the Output Variable names.
- Add the library dependencies that are required for the script to work.
- Write the script code.
- Write the script documentation.
- Test your script by running it with sample data in Testing tab.
- When the script produces the expected output, approve the script.
- Script can be used now in activated data flows.

A script represents a Python function with one parameter, that responds with one value when called.

When the script is called from data flow, the incoming data is passed to the function parameter.

The incoming data and the response can be used in the code by referencing the variable names that are declared for the script.

## Input and output variables names

To process any incoming payload, you must declare the Input Variable and Output Variable names when modeling the script.

When a script executes in runtime, or when the script is tested, the incoming payload is assigned to the Input Variable name you have declared. You must use that variable name in your code to process the incoming payload.

The result of the script must be passed to the variable name that is declared in the Output Variable name field. This is the output generated from the scripting activity step in the data flows.

**Note:** ION Scripting does not validate and check if the declared Input and Output Variables are used in the code.

Input Variable and Output Variable names must conform to Python variable naming convention.

The Input Variable or the Output Variable names can be any length and can consist of uppercase and lowercase letters (A-Z, a-z), digits (0-9), and the underscore character (\_). An additional restriction is that, although a variable name can contain digits, the first character of a variable name cannot be a digit.

The Input Variable and the Output Variable cannot have the same name.

To define the Input Variable and Output Variable names:

- 1 Expand the Scripts right panel
- 2 Specify the Input Variable name.
- 3 Specify the Output Variable name.

## Writing the code

The source-code editor features are designed to simplify and speed up typing of the script code.

These are the editor features:

- Python syntax highlighting
- Indentation
- Keyword autocomplete
- Brace matching functionality
- Line numbering
- Search & Replace

**Note:** ION Scripting does not support syntax validation for programming and stylistic errors in the code.

## Searching text in the code editor

- 1 Click in the editor.
- 2 Press CTRL + F on your keyboard.
- 3 Search bar is displayed.
- 4 Specify the text to search for.  
The matched text is focused and highlighted..
- 5 Use the find next and find previous arrows to find another match.

## Searching and replace text in the code editor

When developing the code, ensure that the Input Variable and Output Variable names are referenced and correctly used in the code.

The libraries that are added as a dependency are installed and their modules can be imported in the script code. See the library's homepage to read the library user guide for instructions on how to use the library.

- 1 Click in the editor.
- 2 Press CTRL + F on your keyboard.  
The search bar is displayed.
- 3 Click the + icon to enable the replace functionality.



- 4 Specify the text to search for and the text to replace it with.
- 5 Click **Replace** to replace the selected match.
- 6 Click **All** to replace all matches in the code.

## Library dependencies

If the code requires to be extended with libraries, you must first upload those libraries. Then add the libraries as a dependency to the script that uses them.

After the libraries are added, you can browse them in the folder structure and open the files contained within the libraries.

If you hover over an added library, you can view these details:

- Which library version is currently used in the script.
- What other versions of the library are available in ION.

To add a library or libraries as a dependency to a script:

- 1 Expand the scripts right panel.
- 2 Expand the Libraries accordion.
- 3 Click the '...' ellipsis menu.
- 4 Click **Add Library**.

The **Select Libraries** dialog box is displayed with the list of all libraries in ION and the versions available for each. For each listed library these details are provided:

- Library name
- Library description
- Available versions
- Last updater's username
- Last updated date

You can search through the libraries using the **Search** field above the data grid. Search is filtering the results according to the library names, library description and user that uploaded the latest version of the library.

- 5 Select one or more rows to add the desired libraries as dependencies to the script.
- 6 Select the version of each library from the Version column if required.
- 7 Click **OK** to add the selected libraries as dependencies to the script.

To fix issues of missing libraries, you can identify the name and version of the missing library and import it in ION Scripting. After the missing library is imported, the error status is removed. If the library dependency is no longer required, you can remove the library from the script.

**Note:** Scripts that have missing libraries as a dependency cannot be approved. You must first fix the error to approve and use this script in active data flows.

---

## Updating a library

To update a library to use the latest version or to use another available version:

- 1 Click the '...' ellipsis menu.
- 2 Click **Change Version**.
- 3 Click the name of the library to be updated.
- 4 Click the version number you wish to use.
- 5 To update all library dependencies to use their latest available versions, click **Update All**.

## Removing a library

To remove a library from the dependencies from the script:

- 1 Click the '...' ellipsis menu.
- 2 Click **Remove Library**.
- 3 Click the name of the library to be removed.
- 4 To remove all libraries from the list of dependencies of the script, click **Remove All**.

## Testing a script

When a test is complete, the output is shown in the Output code pane.

The output can contain this information:

- The expected output payload.
- Python exception.
- Scripting limitation exception.

Executing a test requires that the script is saved. If you have unsaved changes in your script at the moment of testing, a confirmation dialog box is displayed. You are asked to proceed with the test and save the script changes or cancel the test.

When the testing execution completes, the Performance Metrics in the right panel get updated to show:

- The Execution Time: How long the script was running, between 0 and 30 seconds.
- The Input Payload Size: What the size of the input payload was, between 0 and 5 MB.
- The Output Payload Size: What the size of the output payload is.

### Execution time

When testing, the execution is limited to a maximum of 30 seconds. In comparison, the maximum script execution time in runtime is 15 seconds.

You can use the testing and performance metrics to ensure that the scripts runtime execution can fit within the imposed execution time limits.

If the test execution takes over 15 seconds, this is indicated by highlighting the Execution Time in red.

### Input and Output payload size

Script can process and produce payload up to 5 MB in size. The performance metrics for the payloads can help you understand how the script can perform with different payload sizes.

We recommend that you test the script with the largest expected payload. This is to ensure that the Execution Time is within the limit of 15 seconds running time. And that the output that is generated is less than 5MB.

To test a script:

- 1 Switch to the **Testing** tab in the Script details page.
- 2 Paste or upload the input data.
- 3 To upload a document, click the **Upload** icon and select the appropriate file.
- 4 Click **Test** to execute the script with the input data.

## Test data syntax-highlight and pretty-print

When uploading test data, or getting output from running a script, the document's data format is detected. This format is applied to the respective input and output editors.

Data format detection is supported for these types:

- Conventional JSON
- Newline-delimited JSON
- XML

Syntax-highlighting is applied in the editor according to the detected data format type. The data format drop-down list indicates the detected format. If the data format cannot be detected, the selected format in the drop-down list is `Text`. You can manually change the data format type in the drop-down list.

Click the `</>` icon to use formatting (pretty-print) on the documents. The icon is highlighted when the formatting is enabled. Auto-formatting is applied on uploaded and output documents. To disable the formatting of the documents, click the highlighted `</>` icon.

## Script documentation

On the **Documentation** tab, you can write in a rich-text editor to document the functionality of the script and the code.

You can style the text by using the rich-text editor toolbar controls for managing:

- Headers
- Font styling, for example setting the font to bold or changing the color of the text
- Text indentation
- Numbered lists and unordered lists

To save the documentation, save the script. The documentation is exported and imported together with the script.

## Script versioning

You can use the script versioning feature of ION Scripting to version and subsequently update scripts and data flows on the fly.

A script can have multiple versions with each modeled with different code functionality. However, we recommend that you to keep consistency and compatibility between the script versions.

Status' of a script version:

- Draft
- Approved

Initially when a script is created it is in draft status. Draft scripts can be used in model data flows, but they prevent the activation of the flow. To activate the data flow, you must first approve the scripts that are used.

**Note:** Activated data flows use the latest approved version available of the script. When the script is updated with a new approved version, the data flows do not require re-activation. They start using the newly approved version of the script when their last run with the older version completes.

Before a draft version can be approved, it must be valid. The script is considered valid and can be approved, in these situations:

- There is code.
- Input and Output Variable names are declared.
- Added libraries as dependencies do not have an error status.

**Note:** ION Scripting does not support syntax validation for programming and stylistic errors in the code.

In the modeling page of a script, the Versioning widget shows all the available versions. The current open script version is highlighted and in focus. When you open a script from the Overview page, the latest version of the script is displayed. To explore previous versions of a script, click on a different version within the widget.

These details are displayed for a script version in the Versions widget:

- Version number
- Version status
- Last update date and time
- Last updater's username

Click the **Check mark** icon to approve a draft version. This changes the status from draft to approved for the version.

Approved versions are read-only, but you can still change the test input and run the test.

To edit an approved version, you must create a new draft version. A new draft can be created from any of the approved versions available.

On one of the available approved versions, click the **Edit** icon to create a new draft of that version. Each creation of new draft increases the version number. Deleted version numbers are not reused when creating new draft version.

To delete a version, click the **Trash** icon on the version in the Versions widget. Versions that are currently opened cannot be deleted. The delete action is irreversible and saving is not required.

## Using scripting

### Scripts in data flows

Scripts are used as a supporting activity step in Document Flows and Data Lake Flows.

When a scripting step is activated at runtime, the incoming document is sent as a parameter to the code contained in the script for execution.

From the script code perspective, the input document is passed as the Input Variable name; the output document is generated as the value assigned to the Output Variable name.

### Using a script in a data flow

A data flow cannot be activated if it uses a script that has no approved version available.

Activated data flows use the latest approved version available of the script. When the script is updated with a new approved version, the data flows do not require re-activation. They start using the newly approved version of the script when their last run with the older version completes.

To use a script in a data flow:

- 1 Create or open a Document Flow or Data Lake Flow.
- 2 Drag and drop the Scripting activity into the canvas of the flow modeler.
- 3 Click the Scripting widget to assign a script to the step.
- 4 On the **Properties** tab, select a script in the drop-down or create a new one.
- 5 On the **Document Mapping** tab, select the input documents and map each input document to an output document. To use the script, each selected input document must be mapped to an output document.

## Limitations

These are the script runtime limitations:

- Maximum execution time of 15 seconds.
- Scripts can process input payload up to five MB.
- Scripts can output payload up to five MB.
- Script memory limit up to 1000 MB.
- Scripts access to external and online resources is prevented.

A script can only access the script code, the library dependencies attached to it, and the input payload passed at runtime.

## Runtime errors and exceptions

During runtime execution of a script, if an error occurs a Confirm BOD is published because of these reasons:

- Python exception.
- Input Variable name was not used in the code.
- Output Variable name was not used in the code.
- The script does not exist. It was deleted while it was used in an active data flow.
- The input payload exceeds the five MB limit.
- The output payload exceeds the five MB limit.
- The execution exceeded the 15-second runtime limit and timed out.
- The script failed to execute after 10 retries.
- The script failed to execute because of a memory limit.
- The script failed to execute because of reduced networking resources.

## Chapter 7: ION Connect import and export

With ION Desk, you can import and export almost all ION entities.

You can use these import and export options:

- Import ION Desk content that was delivered by Infor.
- Copy ION Desk content from one ION installation to another. For example, to copy your models from a test environment to a production environment.
- Create a backup of your configuration and to restore a previous version of your configuration.
- Import models from a previous ION release.

You can import and export one or more entities such as connection points or mappings. You can also export and import a complete parent entity as data flows including all child connection points and mappings.

### File Formats

The output of an export operation from most ION entities is an XML file. The exception is custom documents definitions where the output is a ZIP file.

### Active / Inactive status

Status information is not included in the import and export procedure. When exporting an active model, this model does not become automatically active after import.

## Import procedure

The import procedure must be initiated on the appropriate page for each entity.

For example, in the **Data Flows** page, you must import the files that are previously created by data flow definitions. You cannot import a file that was created by exporting another entity. For example, you cannot import connection points from the Workflow page. Nonetheless, you can import the connection points from the data flow by importing the data flow import file on the connection point page.

## Override import

When changing a solution that was provided by Infor or a third party, copy the imported items to another name before changing them. This is to avoid overwriting your customization when importing a new version of the same solution.

## Generic import procedure

To import a file containing a specific entity:

- 1 Open the menu by clicking the hamburger icon.
- 2 Navigate to the entity page for the models you want to import.
- 3 Click **Import**. A dialog box where you can locate and select the file opens.
- 4 Select the file to import.
- 5 If the import file contains entities that already exist in ION, you are asked whether to:
  - Skip the already existing models.  
If an entity already exists, it is not imported. Unique entities from the same file are imported normally.
  - Rename duplicated.  
If an entity already exists, the imported entity is renamed so it has a unique name. The name is created by adding the suffix `_#` to the original name, where `#` is a number starting from 1.
- 6 Click **OK**. The entities from the file are imported.  
All imported entities can be customized in the target environment.

## Mapping specific behavior

When the imported file contains single or multiple mapping models that already exist in ION, the generic dialog box is extended with the **Select mappings which will be appended** option.

The duplicated mapping is added to the existing mapping as a new version with the Draft status. Any already existing draft is overwritten by the imported one.

The **Append** action can fail in these situations:

- The imported and existing mappings have different source or target Documents.
- The existing mapping has a legacy status and cannot contain multiple versions.
- The existing mapping has one version, which is a draft currently used in a data flow.

Afterwards you can select an action for the remaining mappings.



## Legacy mapping (without versioning)

When a legacy mapping is imported it is converted into a standard mapping. To convert a legacy mapping that already exists in the system:

- 1 Export the legacy mapping.
- 2 Delete the legacy mapping.
- 3 Import the previously exported mapping and it is handled by the system as a standard mapping.

## Data flow and connection point specific behavior

If the imported file contains entities that already exist. You can be asked to select an action for connection points and other entities separately. Select one of these options

- Skip the already existing.  
If the data flow is unique but some of its referenced entities, for example connection points or mappings, already exist, these entities are skipped. Consequently, the corresponding activities in the imported data flow do not have a connection point or mapping set.
- Rename duplicated.  
If some of its referenced entities already exist, these entities are renamed. The corresponding activities in the data flow refer to the renamed entity.
- Merge additional documents (Connection points only).  
Unique document configurations from already existing connection points are added to the existing connection point.

## Export procedure

To export a file containing a specific entity:

- 1 Open the menu by clicking the hamburger icon.
- 2 Navigate to the entity page for the models you want to export.
- 3 Select the entity to export.
- 4 Click **Export**.

When the not provisioned connection point is selected. You can choose to drop the connection properties when exporting, through the confirmation dialog box. Do not include properties to create a generic delivery to be used in other environments. For exporting connection properties, click **Yes**. For provisioned connection point you cannot export connection point properties.

- 5 Depending on your browser settings, one of these actions happens:
  - A dialog box is displayed to specify a file location and file name for the file. Click **Save**.
  - The file is saved to the default download location with the default file name.

- 6 The selected entity is exported to the file.

**Note: Removing connection points from Data Flows.**

You can remove the connection points from the data flow model but preserve the documents. This is helpful when you must distribute template data flows although the connection points to be used are not known upfront. Go to the data flow detail page and click **Remove connection points**.

## Chapter 8: Event Management

With Event Management you can monitor business events based on pre-defined business conditions and alert users when exceptions take place.

You can monitor these business event types:

- Individual instances of documents  
For example, a new sales order in the form of a `Sync.SalesOrder` document
- Combinations of multiple events  
For example, a sales order and its related shipment
- Non-occurrence of events  
For example, an invoice has never been sent

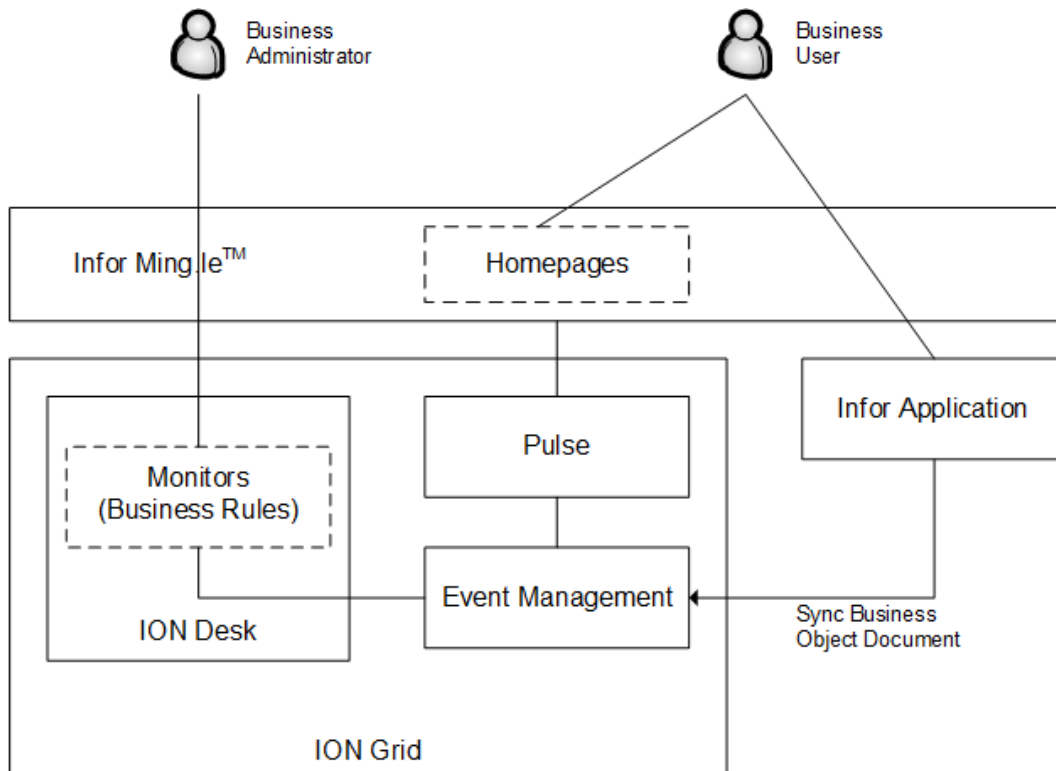
Applications that are enabled to work with ION Connect can use the Event Management functionality without any changes in the application. You can enable monitoring on all documents of type Sync that an application publishes.

To configure business conditions, monitors are defined in ION Desk. A monitor contains a monitoring rule, which consists of one or more conditions. After activation, at runtime, a monitor verifies incoming documents against its monitoring rule. If the result of this evaluation is true, a monitor generates an alert. The alert is sent to a distribution list containing users or user groups. The users receive the alert in their personal Infor Ming.le page, the Infor Inbox application or by email. For details about handling alerts by end users, see these guides:

- *Infor Ming.le User Guide.*
- *Infor Inbox User Guide*

## Architecture

This diagram shows the overall architecture of Event Management:



The Event Management Engine is part of the ION Service installation.. Event Management can work with any document message of type Sync.

When an alert is detected, Event Management creates an alert for the end user.

## Examples

This section contains examples of communication with Application Document messages through the ION Service Bus. A monitor can verify a single document and various documents from different sources.

### Example 1 - Single document Monitor

You define a monitor that verifies incoming Sync.SalesOrder documents. The monitor triggers an alert if a Sync.SalesOrder document contains a status change.

An ERP SalesOrder has status Hold. Inventory arrives and the order status is changed to Open, so the order can be processed. ERP issues an update to this SalesOrder and sends a Sync.SalesOrder document through the Service Bus to Event Management.

In Event Management, the monitor verifies the received Sync.SalesOrder document. Because of the status change, the monitor triggers an alert.

### Example 2 - Monitor with multiple documents

You define a monitor that compares incoming Sync.SalesOrder and Sync.Shipment documents. The monitor triggers an alert if:

- The actual shipment took place later than promised. That is: if the delivery date in the Sync.Shipment document is later than the promised delivery date in the Sync.SalesOrder document.
- A partial shipment was performed. That is: if the delivered quantity in the Sync.Shipment document is less than the quantity in the Sync.SalesOrder document.

In ERP the status of a SalesOrder is changed from Working to Shipped. ERP sends a Sync.SalesOrder document through the Service Bus to Event Management.

The order is shipped. The warehousing application generates a Sync.Shipment document for the SalesOrder. The Status is Shipped. The Sync.Shipment document is sent, through the Service Bus, to ERP and to Event Management.

In Event Management, the monitor compares the received Sync.SalesOrder and Sync.Shipment documents and creates an alert if one of the conditions is met.

## Concepts

### Monitors

A monitor verifies incoming documents against a monitoring rule. If the result of this evaluation is true, the monitor generates an alert. The alert is sent to the users that are based on a distribution list. The distribution list can contain one or more users, distribution groups, contacts, or email addresses. The involved users receive the alert in their personal Infor Ming.le page, the Infor Inbox application, or by email.

For each business domain, you can use a set of monitors to generate alerts relevant to this business area.

A monitor definition contains these logical configuration areas:

- The monitoring part. Here you specify when an alert must be created.
- The alert definition part. Here you specify what should happen with the alert.

For the monitoring configuration, these details are required:

- One or more Application Documents such as `SalesOrder`, which are used for monitoring.
- The monitoring rule to apply for these documents.

A monitoring rule can consist of multiple conditions. For example, for the `SalesOrder` document, you can define a monitor that generates an alert when it is three hours after `SalesOrder.PromisedDeliveryDate`.

See [Monitor rules](#) on page 202.

- A list of the Application Document attributes that is used in the monitoring rule.  
When you create a monitor, you must select these attributes before you define the monitoring rule.
- Occurrence settings that specify how often an alert is generated.  
See [Occurrence settings](#) on page 203.

For the alert configuration, these details are required:

- The alert message and its translations.  
This message is displayed in the Alert summary and used in the e-mail notification sent to the user.  
See "Defining Monitors".
- Optional: Configure the alert due date. The due date is required to configure escalation and reminders to be sent before or after due date.  
See "Defining Monitors".
- Optional: Configure custom drillbacks to be displayed in the alert details. These drillbacks are generated in addition to the standard drillbacks associated with the documents that triggered an alert. See "Defining Monitors".
- Optional: configuration to start a workflow from this alert.  
By default, all alerts can be closed through the **Done** action button. If workflow settings are configured, another action button, **Resolve**, becomes available. With this action, the alert is closed and an instance of the attached workflow is started instantly.  
See "Defining Monitors".
- The distribution list to which the alerts must be sent.  
You can specify to which users, groups, or contacts the alert must be sent. You choose between simple and advanced configuration.  
See "Adding distribution elements".
- The escalation and reminders for this alert.  
Specify escalation rules or reminders for an alert when:
  - The alert must be escalated if it is not assigned in time or if it is not closed in time in a given interval of time since its creation date. You can specify the number of management levels to escalate to.
  - An alert has a due date, you can specify that the alert is escalated, canceled or that a reminder is sent at a given time before or after the due date. You can specify a distribution list for the escalation or the reminder.See "Defining Monitors".

## Application documents

In a monitor you select one or more application documents. If a monitor is active and a Sync message is published for one of the selected documents, that message is sent to the monitor. The monitor uses the data from the BOD to evaluate its rule.

For example, if you select the `SalesOrder` document, any `SyncSalesOrder` message that is published is sent to the monitor.

The monitor remembers the document identifier to use in the alert details for drillback information. The document identifier consists of the document ID, revision ID, accounting entity, and location.

## References

You can use multiple related documents in a monitor. In the monitor definition you can select references from the document metadata defined in the Registry, or use monitor conditions to define the document relations. If you use monitor conditions, you must define valid comparison conditions between the documents and use them in the monitoring rule.

When using monitor conditions to define document relations, you cannot use the monitoring rule of type Document Overdue.

After activation, at runtime, the monitor evaluates if the selected references or the defined conditions are satisfied. Only related documents are used to evaluate the monitor.

**Note:** In a monitor, you cannot use documents that are not related to each other.

## Characteristics of a reference defined as document metadata

Two documents are related if there is a reference from one document to the DocumentID/ID of the other document.

References are defined and stored in the Registry. In Event Management you first choose the documents to be monitored. Then you select the appropriate reference from a list of all possible references between the selected documents. This list is retrieved from the Registry. For details on the Registry, see the *Infor ION Development Guide*.

A reference is defined as a component in a document definition. Examples of reference components are "SalesOrderReference" and "PurchaseOrderReference".

In Event Management this information is displayed for each reference:

- From Application Document
- To Application Document
- From Path

This is the path to the reference in the From Application Document.

If you selected more than two documents for a monitor, the references between these documents must not be circular. For example, if you selected documents A, B, and C, only these references are allowed:

- {A->B->C}
- {A->B and A->C}

Multiple references can exist between two documents. In a monitor, you must select only one reference. See the example.

### Example - Multiple references between two documents

In a monitor definition, you select the SalesOrder and Shipment documents.

This table shows some of the references that are available in the document definitions:

From Application Document	To Application Document	From Path
Shipment	SalesOrder	Shipment/ShipmentItem/SalesOrderReference/ DocumentID
Shipment	SalesOrder	Shipment/ShipmentUnit/ShipmentUnitItem/ SalesOrderReference/DocumentID

In the monitor definition, select only one reference.

## Characteristics of document relations defined as conditions

For each document selected in the monitor, you must define a matching condition to relate this document to one of the other documents used.

The matching conditions must be of type Attribute Comparison. They must use the equals operator and must compare attributes from two different documents.

You must use the matching conditions in the monitor rule by including them in a combined condition used in the rule.

When the monitor is activated, the matching conditions for all documents used in the monitor are validated.

**Note:** Monitors with custom document relations do not support rules of type Document Overdue. Therefore, there is no validation for circular matching conditions.

## Attributes

Attributes are elements from an application document that contain data. Attributes are used in the monitor rule to determine when an alert must be raised.

## Repeating attributes

Some parts of an application document are repeating, which means they exist multiple times. For example, a sales order contains multiple sales order lines. All elements that are included in a repeating component are also repeating. For example, an item code on a sales order line can exist multiple times in a sales order.



If you use repeating attributes in a condition, the condition is met if the application document contains at least one value that meets the condition. For example, you define a rule using this combined condition on a sales order header and sales order line:

```
OrderHeaderStatus = "Planned" and OrderLineItemType = "Production"
```

This means: the status of the sales order header is "Planned", and the order has at least one line having item type "Production".




If you use multiple repeating attributes in a condition, the relation between the attributes is taken into account. For example, you define a rule using this combined condition:



```
OrderLineItemType = "Production" and OrderLineAmount > 100
```

The condition is only met if an order line exists that meets both conditions. For example, an order has two lines, one having type "Production" but amount 50, and another one having amount 200 but type "Purchase". The condition is not met.

## Attribute types

This table shows the attribute types that can be selected in monitor definitions and activation policies.

Attribute type	Icon	Description
Data Element		<p>An element in a business document that can have a value. For example, an Amount element. A data element can be selected.</p> <p><b>Note:</b> If the data element is contained in a repeating group, the data element can exist multiple times in a business document.</p> <p>A data element can contain attributes. To view the attributes, expand the data element node.</p>
Repeating Element		<p>A special type of data element that can exist multiple times in a business document. For example, a Note element.</p> <p>A repeating element can be selected.</p> <p>A repeating element can contain attributes. To view the attributes, expand the repeating element node.</p>
Attribute		<p>An attribute in a business document that can have a value. For example, a currencyID that is linked to an Amount element.</p> <p>An attribute can be selected.</p> <p><b>Note:</b> If the data element is contained in a repeating element or in a repeating group, the attribute can exist multiple times in a business document.</p>

Attribute type	Icon	Description
Group		<p>A grouping element in a business document that does not have a value, but contains other group elements or attributes. For example, a <code>SalesOrderHeader</code>.</p> <p>A group has children, such as data elements, attributes, or other groups. To view the children, expand the group node.</p>
Repeating Group		<p>A special type of group that can exist multiple times in a business document. For example, a <code>SalesOrderLine</code>.</p> <p>A group has children, such as data elements, attributes, or other groups. To view the children, expand the repeating group node.</p>

## Monitor rules

Monitor rules are used to verify incoming documents. Depending on the verification result, alerts can be generated. A rule can consist of multiple conditions.

Event Management supports these types of monitor rules:

- **Condition only**

Use this type with a single document or with multiple documents. The conditions are evaluated if all documents that are involved in the rule occur.

See [Monitor rule - condition only](#) on page 203.

In this type of rule, you can only select one monitor condition. To use multiple conditions, using AND or OR, you must apply a combined condition.

See [Combined condition - Event Management](#) on page 209.
- **Value Change**

Use this type with a single document, for example to monitor status changes within the same document. You can monitor value changes for document attributes of type string or numeric. You can generate alerts in these situations:

  - The value of an attribute changed when comparing two document instances. You can specify a particular initial value and changed value, only the new value, or monitor any change.
  - The value of an attribute did not change in a given interval of time since the first document instance was received. You can specify a particular initial value or monitor for any change.

See [Monitor rule - value change](#) on page 213.
- **Document Overdue**

Use this type of rule with multiple documents, for example to verify if two related documents succeed each other in a given interval of time.

See [Monitor rule - document overdue](#) on page 221.

## Occurrence settings

A monitor definition contains occurrence settings that specify how often an alert is generated.

By default, each monitor is of type "Single Occurrence". An alert is generated each time the evaluation result of the monitor rule is true.

You can also create multiple occurrence monitors. An alert is generated if the monitor rule evaluates to true a specified number of times in a given time interval. For example, you can define a monitor that sends an alert if the monitor rule evaluates to true five times within 24 hours.

## Monitor rule - condition only

A monitor can use comparison conditions based on attributes from the documents that are being monitored. You can also define combined conditions.

Event Management supports these condition types:

- [Attribute-value comparison - Event Management](#) on page 203
- [Attribute comparison](#) on page 204
- [Attribute existence](#) on page 205
- [Attribute comparison with calculation](#) on page 206
- [Attribute comparison with deviation \(%\)](#) on page 207
- [DateTime Check](#) on page 208
- [Combined condition - Event Management](#) on page 209

For details on the condition types, see the corresponding sections.

In monitor conditions you can use various comparison operators and arithmetic operators.

See [Operators in monitor conditions](#) on page 211.

## Attribute-value comparison - Event Management

Use this comparison to compare the value of a document attribute to a fixed value. This type of comparison is useful for attributes whose value does not change frequently.

This table shows the syntax of the attribute-value comparison condition type:

<b>Syntax</b>	[Attribute][Operator][Value]
---------------	------------------------------

<b>Attribute</b>	One of the document attributes selected for monitoring. The attribute can be of type: <ul style="list-style-type: none"> <li>• String</li> <li>• Integer</li> <li>• Decimal</li> <li>• Date and Time</li> <li>• Date</li> <li>• Time</li> <li>• Boolean</li> </ul>
<b>Operator</b>	Comparison operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Value</b>	A constant value of the same type as the attribute.
<b>Evaluation</b>	The evaluation result is True or False.

### Example

You build a condition to monitor contracts that have started after January 1st 2015. In the condition you use the StartTime attribute from the Contract document.

This table shows the condition:

<b>Attribute</b>	StartTime
<b>Operator</b>	>
<b>Value</b>	2015-01-01T00:00:00

The evaluation result of the condition is true if the Date and Time value of the StartTime is after January 1st 2015 00:00 hours.

### Attribute comparison

Use this comparison to compare values of two document attributes that are selected for monitoring. You can compare attributes from the same document or from different documents. The attributes must have the same data type.

This table shows the syntax of the attribute comparison condition type:

<b>Syntax</b>	[Attribute1][Operator][Attribute2]
<b>Attribute 1</b>	One of the document attributes selected for monitoring. The attribute can be of any type.
<b>Operator</b>	Comparison operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.

<b>Attribute 2</b>	One of the document Attributes selected for monitoring. The attribute is of the same type as Attribute 1.
<b>Evaluation</b>	The evaluation result is True or False.

### Example

You build a condition to determine whether a partial shipment has occurred for a sales order line. In the condition you compare the OrderedQuantity attribute from the SalesOrder document with the ShippedQuantity attribute from the Shipment document.

This table shows the condition:

<b>Attribute 1</b>	OrderedQuantity
<b>Operator</b>	>
<b>Attribute 2</b>	ShippedQuantity

The condition evaluates to True if the value of OrderedQuantity is greater than the value of ShippedQuantity.

## Attribute existence

Use this condition type to check if an attribute has a value in an incoming document.

This table shows the syntax of the attribute existence condition type:

<b>Syntax</b>	[ Attribute][Check]
<b>Attribute</b>	One of the document attributes selected for monitoring. The attribute can be of any type.
<b>Check</b>	One of these values: <ul style="list-style-type: none"> <li>Exists the evaluation result is True if the attribute has a value.</li> <li>Does Not Exist the evaluation result is True if the attribute has no value.</li> </ul>
<b>Evaluation</b>	The evaluation result is True or False.

### Example

You build a condition to verify whether the Description attribute is empty.

This table shows the condition:

<b>Attribute</b>	Description
<b>Check</b>	Does Not Exist

The evaluation result of the condition is true if the Description attribute has no value.

## Attribute comparison with calculation

Use this comparison to compare values of two document attributes that are selected for monitoring. You can compare attributes from the same document or from different documents. The attributes must have the same data type.

Use this type of comparison to allow a specific difference between the two attribute values. For example:

- A condition evaluates to True if amount A exceeds amount B with at least 100000.
- A condition evaluates to True if date A is less than 20 days later than date B.

This table shows the syntax of the attribute comparison with calculation condition type:

<b>Syntax</b>	[Attribute1][Operator][Attribute2][Operator][Value]
<b>Attribute 1</b>	One of the document attributes selected for monitoring. The attribute can be of type: <ul style="list-style-type: none"> <li>• Integer</li> <li>• Decimal</li> <li>• Date and Time</li> </ul>
<b>Operator</b>	Comparison operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Attribute 2</b>	One of the document Attributes selected for monitoring. The attribute is of the same type as Attribute 1.
<b>Operator</b>	Arithmetic operator; depends on the attribute type. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Value</b>	A constant value of the same type as the attributes.
<b>Evaluation</b>	The value of Attribute 1 is compared with the result of the calculation between Attribute 2 and the specified Value. The evaluation result is True or False.

### Example

You build a condition to verify whether the duration of a contract is less than 20 days. In the condition you compare the StartTime and EndTime attributes of the Contract document.

This table shows the condition:

<b>Attribute 1</b>	EndTime
<b>Operator</b>	<
<b>Attribute 2</b>	StartTime
<b>Operator</b>	+

<b>Value</b>	20 days
--------------	---------

## Attribute comparison with deviation (%)

Use this comparison to compare values of two document attributes that are selected for monitoring. You can compare attributes from the same document or from different documents. The attributes must have the same data type.

Use this type of comparison to allow a specific deviation percentage between the two attribute values. For example, you can create a condition that evaluates to True if amount A deviates more than 5 % from amount B.

This table shows the syntax of the attribute comparison with deviation (%) condition type:

<b>Syntax</b>	[Validate][Deviation percentage][From]
<b>Validate</b>	One of the document attributes selected for monitoring. The attribute can be of type Integer or Decimal. This is the value to be verified.
<b>Deviation percentage</b>	A percentage (between 1% and 99%) that is calculated from Attribute 2.
<b>From</b>	One of the document Attributes selected for monitoring. The attribute is of the same type as Attribute 1. This is the value used as reference.
<b>Evaluation</b>	First, the allowed deviation is calculated as: $D = (\text{Deviation\%} / 100) * \text{Attribute2}$ Then Event Management verifies whether the value of Attribute 1 is less than the allowed deviation ( $\text{Attribute 1} < \text{Attribute 2} - D$ ) or greater than the allowed deviation ( $\text{Attribute 1} > \text{Attribute 2} + D$ ). The evaluation result is True or False.

### Example

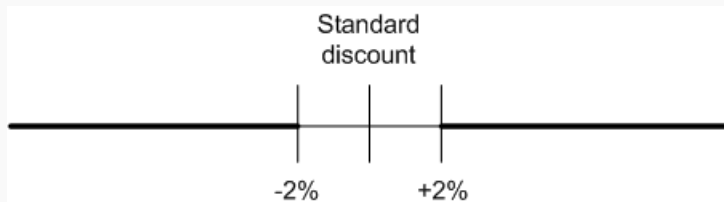
You build a condition to verify that each discount given for a Sales Order Line deviates more than two percent from the standard discount on the sales order header.

This table shows the condition:

<b>Validate (Attribute 1)</b>	LineDiscount
<b>Deviates (Deviation%)</b>	2%

<b>From (Attribute 2)</b>	HeaderDiscount
---------------------------	----------------

The condition evaluates to true for the values beyond the deviation percentage. This diagram shows these values as a bold line.



## DateTime Check

Use this type of condition to generate an alert at a specified moment in time. This moment is calculated relative to an attribute of type Date and Time, which is selected for monitoring.

This table shows the syntax of the DateTime Check condition type:

<b>Syntax</b>	[Time Offset][Operator][DateTime Attribute]
<b>Time Offset</b>	A given time period expressed in days, hours, minutes, or seconds.
<b>Operator</b>	Comparison operator: before or after. See <a href="#">Operators in monitor conditions</a> on page 211.
<b>Attribute</b>	One of the document attributes selected for monitoring. The attribute is of type Date and Time.
<b>Evaluation</b>	Evaluates to True when the calculated point in time has passed.

### Example

You want to receive an alert eight hours before the time stored in the ExpirationDate attribute of the SalesOrder document.

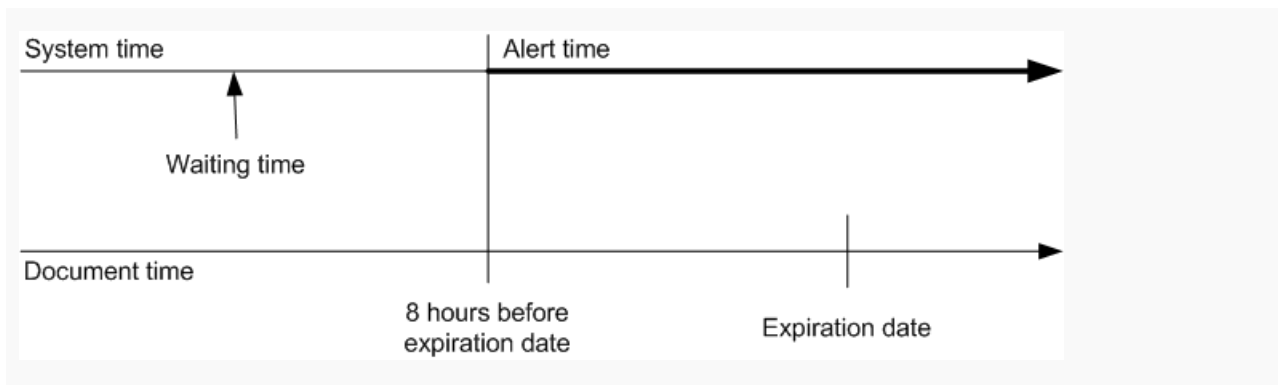
This table shows the condition you create:

<b>Time Offset</b>	8 hours
<b>Operator</b>	before
<b>Attribute</b>	ExpirationDate

The condition evaluates to true if the calculated time is in the past. Otherwise the monitor waits until the system time becomes the calculated time and then sends an alert.

This diagram shows a graphical representation of the condition:





## Combined condition - Event Management

Create a combined condition to evaluate several condition types simultaneously in the monitor rule. All condition types can be combined using the logical operators AND and OR.

The evaluation result of a combined condition is based on the evaluation results of the sub-conditions. Parentheses are used to specify the order of evaluation.

This table shows the syntax of a combined condition:

<b>Syntax</b>	[Condition1] AND   OR [Condition2] <b>Note:</b> You can have more than two sub conditions in one combined condition. For example: [Condition1] AND [Condition2] AND [Condition3]
<b>Condition1</b>	A monitor condition already defined in the list of conditions.
<b>Condition2</b>	A monitor condition already defined in the list of conditions.
<b>Evaluation if AND is used</b>	Evaluates to True if all conditions are True.
<b>Evaluation if OR is used</b>	Evaluates to True if at least one of the conditions is True.

This table shows the syntax that you can use to nest combined conditions:

<b>Syntax</b>	[Condition1] AND   OR ([Combined Condition])
<b>Condition1</b>	A monitor condition already defined in the list of conditions.
<b>Combined Condition</b>	A monitor combined condition, already built using AND or OR, based on the list of conditions. Combined conditions that are reused are surrounded by round parenthesis.
<b>Evaluation</b>	The comparison conditions are evaluated first. Then the combined conditions marked by parenthesis are evaluated. The evaluation result is True or False.

**Note:** If one of the sub-conditions is of type DateTime Check, the evaluation of the combined condition waits for the timer to expire.

### Example 1

You want to generate an alert if a sales order is shipped, and the shipment is late or it was partially shipped. You define a monitor that evaluates the SalesOrder and Shipment documents.

The SalesOrder and Shipment documents are related. In the monitor definition, you must select a reference between these Application Documents.

See [References](#) on page 199.

This table shows the conditions you create:

Name	Type	Condition	Description
SalesOrderShipped	Attribute-Value Comparison	SOStatus = Shipped	Checks whether a sales order is shipped.
PartialShipment	Attribute Comparison	ShippedQuantity < OrderedQuantity	Checks whether only a part of the ordered items were shipped.
DelayedShipment	Attribute Comparison	PromisedShipDate-Time < ActualShipdate-Time	Checks whether the shipment was late.
PartialOrDelayedShipment	Combined Condition	SalesOrderShipped AND (PartialShipment OR DelayedShipment)	Checks whether a partial or delayed shipment took place.

You define a monitor rule of type Condition Only. In this rule, you select the PartialOrDelayedShipment combined condition. When a sales order with status shipped is intercepted, the monitor starts waiting until it receives the corresponding shipment document. When the sales order and shipment documents are both available, the monitor compares the attributes from the defined conditions. If the combined condition evaluates to true, an Alert is sent.

### Example 2

You want to generate an alert if a partial shipment takes place for items in the sales order. You define a monitor that evaluates the SalesOrder and Shipment documents.

The SalesOrder and Shipment documents are related. In the monitor definition, you must select a reference between these Application Documents.

See [References](#) on page 199.

This table shows the conditions you create:

Name	Type	Condition	Description
SameLine	Attribute Comparison	ShipmentRefLineNr = SOLineNr	Ensures data is read from the shipment that belongs to the order line.

Name	Type	Condition	Description
SameItem	Attribute Comparison	ShipmentItemID = SO-LineItemID	Ensures shipment data is read for the item used in the order line.
ShipmentOccurs	Attribute-Value Comparison	ShipmentStatus = Shipped	Checks whether items were shipped.
LessQuantity	Attribute Comparison	ShippedQuantity < SO-LineQuantity	Checks whether only a part of the ordered items were shipped.
PartialShipment	Combined	ShipmentOccurs AND SameLine AND SameItem AND LessQuantity	Checks whether a partial shipment took place.

You define a monitor rule of type Condition Only. In this rule, you select the PartialShipment combined condition. When a sales order with status shipped is intercepted, the monitor starts waiting until it receives the corresponding shipment document. When the sales order and shipment documents are both available, the monitor compares the attributes from the defined conditions. If the combined condition evaluates to true, an Alert is sent.

For more information, see [Defining combined conditions](#) on page 233.

## Operators in monitor conditions

In monitor conditions, you can use these operators:

- Comparison operators
- Arithmetic operators

This table shows the comparison operators per data type:

Attribute Data Type	Operator	Description
All attribute types (in Attribute existence condition)	Exists	True if attribute has a value.
	Does Not Exist	True if attribute has no value.
Boolean	=	True if values match.
	<>	True if values do not match.

Attribute Data Type	Operator	Description
String	=	True if strings match (case sensitive).
	<>	True if strings do not match (case sensitive).
	starts with	True if the attribute starts with the specified string value.
	does not start with	True if the attribute does not start with the specified string value.
	contains	True if the attribute contains the specified string value.
	does not contain	True if the attribute does not contain the specified string value.
	ends with	True if the attribute ends with the specified string value.
	does not end with	True if the attribute does end with the specified string value.
	in set	True if the attribute is equal to one of the values that are specified in the Set. Set is a list of string values separated by commas.
	not in set	True if the attribute is not equal to any of the values from the set. Set is a list of string values separated by commas.
	in codes	True if the attribute is equal to one of the codes that are selected from a specified codelist.
not in codes	True if the attribute is not equal to any of the values that are selected from a specified codelist.	
Numeric, Duration	=	Equals
	<	Less than
	<=	Less than or equal
	>	Greater than
	>=	Greater than or equal
	<>	Does not equal
Numeric	between	True if the value of the attribute is between two specified numeric values. This operator can be used in Attribute Value Comparison condition.
DateAndTime (in Date/Time Check Comparison)	before	Before
	after	After

Attribute Data Type	Operator	Description
DateAndTime, Date	=	Equals
	<>	Does not equal
	<	Before
	<=	Before or equal to
	>	After
	>=	After or equal to
	between	True if the attribute value is between two specified values. This operator can be used in Attribute Value Comparison condition

This table shows the arithmetic operators:

Attribute Data Type	Operator	Calculation
Numeric	+	Addition
	-	Subtraction
	*	Multiplication
	/	Division
DateAndTime, Date	+	Time in the future relative to attribute value
	-	Time in the past relative to attribute value

## Monitor rule - value change

Use this type of rule to monitor changes in the values of attributes in a specified document over a period of time. You can monitor if a value changes to, or does not change to, a different value. A starting and ending value is specified. Intermediate value changes are possible and might influence the monitor result.

This type of monitoring rule evaluates instances of the same document by verifying the tenant ID and the document ID.

### Note:

- You can only monitor attributes of data types String, Numeric, Date, and DateAndTime. The Date and DateAndTime values from the messages are compared with the values in the monitor rule. The selected Date and DateAndTime values are displayed as local date and time.
- You cannot use an attribute that is repeating or an attribute, which has a parent element that is repeating. If such attributes are selected as the value change attributes, an error is given upon activation of the monitor.

### Caution:

- You can select a condition in the value change rule. The rule is applied only for documents for which the condition is True.
- For Value Change monitoring you should use only conditions that contain attributes of the same document.
- Do not use a condition that contains the Attribute that was selected for the value change rule. Otherwise the result is unpredictable. If you combine a value change rule with a condition, the condition is always applied to the last available instance of the business object that is used to evaluate the rule.

The Value Change rule can be built using one of these operators:

- **Changes**  
Use this operator to verify if an attribute value changes from any value to any other value.
- **Changes from any value to new value**  
Use this operator to verify if an attribute value is or becomes the specified value, irrespective of the interval of time.
- **Changes from old value to new value**  
Use this operator to verify if an attribute value changes from a given initial value into a different specified value, or if it changes from a given initial value to a different value and then into the specified new value, within a given period of time.
- **Does not change**  
Use this operator to verify if an attribute value does not change within a given period of time.
- **Does not change from old value to new value**  
Use this operator to verify if a Status attribute does not change from a given initial value into the expected new value, within a given period of time.

Depending on the selected operator, the behavior of the value change rule can be explained in a different way. Therefore, each variation of the rule is explained in detail below.

## 'Changes' operator

This table shows the syntax for rules that use the 'Changes' operator:

<b>Syntax</b>	[Attribute] [Operator] [optional Condition]
<b>Attribute</b>	One of the document attributes selected for monitoring. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Operator</b>	Changes
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. See <a href="#">Monitor rule - value change</a> on page 213.

<b>Evaluation if Operator = Changes</b>	Evaluates to true if an instance of the document was received followed by a new instance of the same document, with a different value than the previous one.  If a Condition is specified, only the document instances for which the Condition evaluates to True are considered for evaluation.
-----------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

This table shows when alerts are generated:

Document instances	Alert is generated
First document instance contains the Attribute with any value.	No
Two document instances follow each other, each having a different value for the Attribute.	Yes
Two document instances follow each other, both having the same value for the Attribute.	No

**Note:** If several document instances are received with the same CreationDateTime, but different values for the Attribute, only one of the document instances is used in the evaluation. No alert is generated until a new document instance with a different CreationDateTime and a different attribute value is received.

### Example - Changes

You can make changes to a planned Shipment. For example, another carrier is assigned. If this is a change to be alerted on, you can create a rule to monitor value changes on the carrier party from the Shipment document.

This table shows the syntax of the rule:

<b>Attribute</b>	CarrierParty
<b>Old Value</b>	N/a
<b>Operator</b>	Changes
<b>New Value</b>	N/a
<b>Within</b>	N/a

### 'Changes from any value to new value' operator

This table shows the syntax for rules that use the 'Changes from any value to new value' operator:

<b>Syntax</b>	[Attribute] [Operator] [New Value] [optional Condition]
<b>Attribute</b>	One of the document attributes selected for monitoring. See <a href="#">Monitor rule - value change</a> on page 213.

<b>Operator</b>	Changes from any value to new value
<b>New Value</b>	A constant value for the attribute that is monitored.
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Evaluation if Operator = Changes from any value to new value</b>	<p>Evaluates to True if one of these conditions is met:</p> <ul style="list-style-type: none"> <li>An instance of the document with [New Value] has occurred.</li> <li>An instance of the document with a different value than [New Value] has occurred, followed by an instance of the document with the [New Value] for the attribute monitored.</li> </ul> <p>If a Condition is specified, only the document instances for which the Condition evaluates to True are considered for evaluation.</p> <p>There is no time limit for the evaluation of this monitor. In case of repeated values changes, the monitor creates an alert each time the Attribute value becomes the New Value.</p>

This table shows when alerts are generated:

Document instances	Alert is generated
First document instance contains the Attribute with the specified [New Value].	Yes
First document instance with Attribute value [New Value] directly followed by another document instance with Attribute value [New Value].	Yes (see also Note below)
First document instance with Attribute value [New Value] directly followed by another document instance with a different Attribute value.	Yes
Document instance with a different Attribute value directly followed by another document instance with Attribute value [New Value].	Yes
Several document instances with a different Attribute value followed by a document instance with Attribute value [New Value].	Yes
Several document instances follow each other with the attribute value alternating any value and [New Value] at different moments in time.	Yes (alert is generated each time the Attribute values becomes [New Value])

**Note:** If several document instances are received with the same CreationDateTime and the Attribute value [New Value], an alert is generated for each document instance. If several document instances are received following each other, but with different Creation Date Time and the Attribute value [New Value], then only one alert is generated.

#### Example - Changes from any value to new value

Sales orders can be "blocked" for various reasons during processing. If you do not want to monitor for a specific status transition, but only want to identify when a sales order has become blocked, you model a rule to check for a status change from any value to value "blocked".



This table shows the syntax of the rule:

<b>Attribute</b>	Status
<b>Old Value</b>	N/a
<b>Operator</b>	Changes from any value to new value
<b>New Value</b>	Blocked
<b>Within</b>	N/a

### 'Changes from old value to new value' operator

This table shows the syntax for rules that use the 'Changes from old value to new value' operator:

<b>Syntax</b>	[Attribute] [Old Value] [Operator] [New Value] [Within] [optional Condition]
<b>Attribute</b>	One of the document attributes selected for monitoring. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Old Value</b>	A constant value for the attribute that is monitored.
<b>Operator</b>	Changes from old value to new value
<b>New Value</b>	A constant value for the attribute that is monitored.
<b>Within</b>	Waiting time the monitor must evaluate value change occurrences since the document with the attribute value [Old Value] has been received by ION.
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Evaluation if Operator = Changes from old value to new value</b>	<p>Evaluates to True if one of these conditions is met:</p> <ul style="list-style-type: none"> <li>An instance of the document has occurred with the attribute value equal to [Old Value] directly followed by an instance of the document with the attribute value equal to [New Value] (without observing a time interval).</li> <li>An instance of the document has occurred, with the attribute value equal to [Old Value], followed by an instance of the document with a different attribute value. The arrival of the second document is the start of the time interval. If an instance of the document with the attribute value [New Value] follows before the timer expires, the rule evaluates to True.</li> </ul> <p>In all other situations, the rule evaluates to False.</p>

This table shows when alerts are generated:

Document instances	Alert is generated
Document instance with the attribute value equal to [Old Value], directly followed by an instance of the document with the attribute value equal to [New Value].	Yes
Document instance with the attribute value equal to [Old Value], followed by one or more instances of the document with a different value, and followed by a document instance with [New Value] within the time interval.	Yes if the document instance with [New Value] occurs before the timer expires. No if the document instance with [New Value] occurs after the timer expires.

**Note:**

- The start of the time interval is the moment the first change from [Old Value] to a different value is detected.
- Occurrence of document instances is evaluated as they enter ION, not based on the document Creation Date Time.
- A deviation of one minute for the time interval duration might occur based on the internal polling interval of the event management engine.

**Example - Changes from old value to new value**

You build a rule to verify if a new sales order is canceled within eight hours after its creation. The monitor will trigger if the sales order status changed from Open to any other status, and then the order was canceled within eight hours, or when the order was Open and then Cancelled. In the rule, you use the **Status** attribute from the SalesOrder document.

This table shows the syntax of the rule:

<b>Attribute</b>	Status
<b>Old Value</b>	Open
<b>Operator</b>	Changes from old value to new value
<b>New Value</b>	Canceled
<b>Within</b>	8 hours

**'Does not change' operator**

This table shows the syntax for rules that use the 'Does not change' operator:

<b>Syntax</b>	[Attribute] [Operator] [Within] [optional Condition]
<b>Attribute</b>	One of the document attributes selected for monitoring. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Operator</b>	Does not change

<b>Within</b>	Waiting time the monitor must evaluate value change occurrences since the first instance of the document occurred after the monitor was activated.
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Does not change</b>	Evaluates to true if these conditions are both met at the end of the time interval: <ul style="list-style-type: none"> <li>An instance of the document was received.</li> <li>No new instance of the same document, with a different attribute value than the previous one, was received within the specified waiting time interval after the first instance.</li> </ul>

This table shows when alerts are generated:

<b>Document instances</b>	<b>Alert is generated</b>
One document instance contains the Attribute with any value.	Yes (after the timer expires)
Two document instances follow each other within the given interval of time, each having a different value for the Attribute.	No (relative to the first document)
Two document instances follow each other within the given interval of time, both having the same value for the Attribute.	Yes (after the timer expires)
One document instance contains the Attribute with any value, followed by another document instance after the timer expires.	Yes (after the timer expires)

#### Example - Does not change

Sometimes you want to ensure that certain documents are handled in a short interval of time.

For example, you create a rule to monitor if the status of an Invoice does not change in 30 days. You do not want to be alerted if the Invoice is closed.

This table shows the syntax of the rule:

<b>Attribute</b>	InvoiceStatus
<b>Old Value</b>	N/a
<b>Operator</b>	Does not change
<b>New Value</b>	N/a
<b>Within</b>	30 days
<b>Condition</b>	Invoice Status <> Closed

## 'Does not change from old value to new value' operator

This table shows the syntax for rules that use the 'Does not change from old value to new value' operator:

<b>Syntax</b>	[Attribute] [Old Value] [Operator] [New Value] [Within] [optional Condition]
<b>Attribute</b>	One of the document attributes selected for monitoring. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Old Value</b>	A constant value for the attribute that is monitored.
<b>Operator</b>	Does not change from old value to new value
<b>New Value</b>	A constant value for the attribute that is monitored.
<b>Within</b>	Waiting time the monitor must evaluate value change occurrences since the document with the attribute value [Old Value] has occurred.
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. See <a href="#">Monitor rule - value change</a> on page 213.
<b>Evaluation if Operator = Does not change from old value to new value</b>	Evaluates to True if these conditions are both met at the end of the time interval: <ul style="list-style-type: none"> <li>An instance of the document with [Old Value] has occurred.</li> <li>No instance of the document with [New Value] was received by the time the timer offset has expired.</li> </ul>

This table shows when alerts are generated:

<b>Document instances</b>	<b>Alert is generated</b>
One document instance contains the Attribute with the [Old Value].	Yes (after the timer expired)
Document instance with the attribute value equal to [Old Value] followed by an instance of the document with the attribute value equal to [New Value].	No if the document with the [New Value] is received within the time interval. Yes if the document with the [New Value] is received after the expiration of the time interval.
Document instance with the attribute value equal to [Old Value] followed by an instance of the document with the attribute value different than the [New Value].	Yes
Document instance with the attribute value equal to [Old Value] followed by an instance of the document with a different attribute value and then followed by a document instance with the attribute value equal to [New Value].	No if the document with the [New Value] is received within the time interval. Yes if the document with the [New Value] is received after the expiration of the time interval.

### Example - Does not change from old value to new value

You build a rule to verify if a new sales order is not picked up within 8 hours after its creation. In the rule, you use the **Status** attribute from the SalesOrder document.

This table shows the syntax of the rule:

<b>Attribute</b>	Status
<b>Old Value</b>	Open
<b>Operator</b>	Does not change from old value to new value
<b>New Value</b>	Working
<b>Within</b>	8 hours

## Monitor rule - document overdue

Use this monitor rule to verify if two related documents succeed each other. The time interval to monitor starts at one of these moments:

- The moment the first document is sent
- The Date and Time value from a specified attribute of the first document

The duration of the time interval is specified in a time offset.

This table shows the syntax of the Document Overdue rule:

<b>Syntax</b>	{[Reference Document][optional Condition]}.{[Time Offset] [Since]}.{[Overdue Document][optional Condition]}
<b>Reference Document</b>	The first document to be monitored.
<b>Condition (optional)</b>	A monitor condition from the list of predefined conditions. The condition must only contain attributes of the first document (the reference document). The monitor only runs for documents for which the condition is True.
<b>Time offset</b>	The time the monitor must wait for the occurrence of the second document (the Overdue Document). The time interval to monitor starts at one of these moments: <ul style="list-style-type: none"> <li>• The moment the first document is sent, expressed as CreationDate-Time in the document header</li> <li>• The time in the [Date and Time Attribute] in the first document if this was specified</li> </ul>

<b>Since (optional)</b>	<p>A Date and Time attribute from the reference document. This attribute has been pre-selected in the list of document attributes to monitor. The time value in this attribute marks the beginning of the monitoring time interval. If the date is in the past, the timer starts immediately. The monitor waits for the related document until the time offset expires.</p> <p>If no attribute is specified, the time interval to monitor starts at the moment the first document is sent.</p>
<b>Overdue Document</b>	<p>The second document, expected to occur within the time offset specified. The overdue document must be related to the first document through a reference. You must pre-select this reference in the monitor references.</p>
<b>Condition (optional)</b>	<p>A monitor condition from the list of predefined conditions.</p> <p>In this condition, you can use attributes from the document that is selected as the overdue document.</p> <p>The monitor evaluates this condition for each document that is related to the reference document. The monitor ignores documents for which this condition is false.</p>
<b>Evaluation</b>	<p>The evaluation starts when an instance of the reference document is received by ION.</p> <p>If a condition was specified for this document, the evaluation starts only if the condition evaluates to True.</p> <p>When the reference document is received, the monitoring time interval is calculated. The interval starts at one of these moments:</p> <ul style="list-style-type: none"> <li>• The moment the first document is created by the source application (CreationDateTime)</li> <li>• The time in the [DateTime Attribute].</li> </ul> <p>The interval ends after the specified time offset.</p> <p>During the specified waiting time, the monitor verifies if an instance of the overdue document occurs. This document instance must be related to the reference document instance.</p> <p>If the overdue document arrives within the time interval, first the associated condition is evaluated.</p> <p>If this condition evaluates to True, or if no condition was specified, the monitor rule evaluates to False.</p> <p>If the condition evaluates to False, the monitor keeps running until the time offset expires.</p> <p>If the time offset expires and no valid occurrence of the overdue document is detected, the rule evaluates to True and an alert is sent.</p> <p>So the rule evaluates to True when the related document is too late or when this document never arrives. The rule evaluates to False if the document is in time.</p>

### Example 1 WITHOUT ATTRIBUTE

You want to verify whether an invoice is issued within 24 hours after a shipment. To achieve this, you define a monitor rule of type Document Overdue for the SalesOrder and Invoice documents. These two documents are related to each other through the reference from Invoice/InvoiceLine/SalesOrderReference/DocumentID/ID to the SalesOrder Document ID.

You are only interested in SalesOrder documents with status Shipped. Therefore, you use a condition of type Attribute-Value Comparison for the reference document.

This table shows the syntax of the rule:

<b>Reference Document</b>	SalesOrder
<b>Condition</b>	SalesOrder.Status = 'Shipped'
<b>Time Offset</b>	24 hours
<b>Since</b>	<empty>
<b>Overdue Document</b>	Invoice
<b>Condition</b>	None

### Example 2 WITH ATTRIBUTE

You want to monitor if a Shipment document for a specific product occurs within eight hours from the shipped sales order's **Promised Delivery Date**. To achieve this, you define a monitor rule of type Document Overdue with Attribute.

You want to monitor only sales orders with status Planned and Shipments with status Shipped. To check the status of the documents, you use conditions of type Attribute-Value Comparison.

The Shipment and SalesOrder documents are related to each other. The relation is based on references from the Shipment to the SalesOrder Document ID. For example, you can use the reference from the Shipment/ShipmentItem/SalesOrderReference/DocumentID/ID to the SalesOrder Document ID.

This table shows the syntax of the rule:

<b>Reference Document</b>	SalesOrder
<b>Condition</b>	SalesOrder.Status = 'Planned'
<b>Time Offset</b>	8 hours
<b>Since</b>	SalesOrder.PromisedDeliveryDate
<b>Overdue Document</b>	Shipment
<b>Condition</b>	Shipment.Status = 'Shipped'

---

# Procedures

## Task overview

In Event Management, you can perform these tasks:

- **Configure monitors**  
You can define new monitors for Event Management or import predefined monitors. See these sections:
  - [Defining monitors](#)
  - [Importing monitors](#) on page 236
- **Activate monitors**  
After activation, a monitor starts evaluating incoming Sync documents from the service bus. See [Activating monitors](#) on page 234.
- **Temporarily suspend active monitors**  
When you suspend a monitor, the data recorded by the monitor is kept and is processed when the monitor is resumed. See these sections:
  - [Pausing active monitors](#) on page 402
  - [Resuming monitors](#) on page 402
- **Change active monitors**  
You can change some settings of an active monitor without deactivating the monitor. See [Changing active monitors](#) on page 235.
- **Temporarily deactivate monitors**  
In case of major changes you must deactivate a monitor before you can change the monitor definition. When you deactivate a monitor, the data recorded by the monitor is removed. See these sections:
  - [Deactivating monitors](#) on page 235
  - [Activating monitors](#) on page 234
- **Deactivate monitors**  
If a monitor is no longer required, you can deactivate it. See [Deactivating monitors](#) on page 235.
- **Export monitors to XML files and import monitors from XML files**  
See these sections:
  - [Exporting monitors](#) on page 235
  - [Importing monitors](#) on page 236
- **Show alerts from an active monitor**  
See [Showing alerts](#) on page 402.
- **Show triggers processed by an active monitor**  
See these sections:



- [Showing triggers](#) on page 403
- [Viewing triggers](#) on page 403
- View archived monitors  
See [Viewing archived monitors](#) on page 404.

All tasks are performed in ION Desk.

## Modeling tasks

### Defining monitors

#### Procedure summary

- 1 Specify the name and description of the new monitor.
- 2 Select the documents the monitor must evaluate.
- 3 If you selected multiple documents, specify references.
- 4 Select document attributes.
- 5 Optionally, specify filters for selected attributes.
- 6 Optionally, create conditions.
- 7 Specify the monitoring rule.
- 8 Specify the alert message.
- 9 Optionally, specify the alert due date.
- 10 Specify distribution details.
- 11 Optionally: specify drillbacks.
- 12 Optionally, specify workflow settings.
- 13 Optionally, specify escalation and reminders details.
- 14 Save the monitor.

#### Procedure details

- 1 Specify the name and description of the new monitor.

Complete these steps:

- a In the **Monitors** modeling page, click **Add** to open a new **Monitor** detail page.
- b Specify this information:

##### **Name**

Specify a unique name that identifies the monitor. The name must contain the characters a-z, A-z, 0-9 and ‘\_’ (underscore). The maximum length is 255 characters. Multi-byte characters are not supported in the name of a monitor.

**Description**

Specify the description of the monitor. The description is used only inside ION Desk, to describe the purpose of the monitor to the business administrator.

**2** Select the documents the monitor must evaluate.

a In the **Documents** tab in the **Monitor** detail page, click **Add** to display the **Add Documents** window.

b Select the document(s) to be monitored.

To filter the list of documents:

- Select a type, when applicable, from the Type list.
- Select a level, when applicable, from the Level list.
- Specify a text in the **Filter** field.

If you have selected the document type "BOD", you can also filter by document level "Custom" and "Standard". Filtering by level is not applicable for the types "JSON" and "ANY", as the documents of these types are always custom.

c Click **OK** to add the selected documents to the monitor.

**3** If you selected multiple documents, specify references.

If multiple documents are selected, specify how these documents refer to each other. Documents that are not logically linked through a reference cannot be used in the same monitor.

Complete these steps:

a Expand the **References** section of the **Documents** tab and click **Add** to display the **Select References** window.

b Select the references to be used in the monitor.

Points of attention:

- Multiple references can exist between two documents. Select only 1 reference.
- If you selected more than two documents for a monitor, the references between these documents must not be circular. For example, if you selected documents A, B, and C, only one of these possible references is allowed:
  - {A->B->C}
  - {A->B and A->C}

See [References](#) on page 199.

c Click **OK**. The selected references are displayed in the **References** section.

d Click **Save**.

**Note:** To use monitor conditions to define document relations, select the **Use conditions to define document relations** check box and proceed with the rest of the monitor configuration. When you define the monitor conditions, define document relations based on Attribute Comparison conditions, using the equals operator, between attributes of monitored documents.

See [References](#) on page 199.

**4** Select document attributes.






Select at least one attribute of the monitor's documents. You can use the selected attributes in the monitor rule, the monitor conditions, or the alert message.

Complete these steps:

a In the **Attributes** tab, click **Add** to display the **Select Attributes** window.

- b Expand the tree and select the desired attributes. The selected attributes are displayed in the tooltip of the “Selected” count at the top of the window.

This table shows the attribute types that can be selected in the **Select Attributes** window.

Attribute type	Icon	Description
Data Element		<p>An element in a business document that can have a value. For example, an Amount element. A data element can be selected.</p> <p><b>Note:</b> If the data element is contained in a repeating group, the data element can exist multiple times in a business document.</p> <p>A data element can contain attributes. To view the attributes, expand the data element node.</p>
Repeating Element		<p>A special type of data element that can exist multiple times in a business document. For example, a Note element.</p> <p>A repeating element can be selected.</p> <p>A repeating element can contain attributes. To view the attributes, expand the repeating element node.</p>
Attribute		<p>An attribute in a business document that can have a value. For example, a currencyID that is linked to an Amount element.</p> <p>An attribute can be selected.</p> <p><b>Note:</b> If the data element is contained in a repeating element or in a repeating group, the attribute can exist multiple times in a business document.</p>
Group		<p>A grouping element in a business document that does not have a value, but contains other group elements or attributes. For example, a SalesOrderHeader.</p> <p>A group has children, such as data elements, attributes, or other groups. To view the children, expand the group node.</p>
Repeating Group		<p>A special type of group that can exist multiple times in a business document. For example, a SalesOrderLine.</p> <p>A group has children, such as data elements, attributes, or other groups. To view the children, expand the repeating group node.</p>

- c Click **OK**. The selected attributes are displayed in the **Attributes** tab.
- d Optionally, change the names of the attributes in the **Attributes** tab. For example, you can specify the name of the document, to which the attributes belong, as a prefix. This is useful if you selected attributes from multiple documents. The attribute names are used as labels for the alert details when they are displayed to the end user.

**Note:**

- You can select these predefined attributes from the document header information:
  - AccountingEntityID

- LocationID
- actionCode

Use these attributes to model that the generation of alerts must depend on the document characteristics instead of on the business data.

For example: You want to monitor that alerts are only sent for documents coming from a specific accounting entity. Therefore, you include a comparison condition based on the AccountingEntityID attribute in the monitor.

- The actionCode attribute can have these predefined values:
  - 'Add'
  - 'Change'
  - 'Delete'
  - 'Replace'

To model that alerts are created only for new records, include this comparison condition in the monitor: `actionCode=Add`

## 5 Optionally, specify filters for selected attributes.

Data Elements and Attributes that are part of a repetitive structure can require additional filtering to be used in a condition. For example, the Location from the ShipToParty information on a Contract document may exist several times. To specify what Location address details to use in the condition evaluation, add a filter on the attribute "type" of the Location element.

To specify a filter:

- In the **Attributes** tab, add one or more attributes that belong to a repeating group. For example: Contract/ContractHeader/ShipToParty/Location/Address/StreetName.
- Select the attribute. The **Filter** button becomes enabled.
- Click **Filter** to display the **Edit Filter** window.
- The xml-attributes of the selected element and the xml-attributes of its repeating parent elements are displayed. Specify the attribute to use for filtering. For example, specify **type**.
- Click in the Equals Value column and specify a value. For example, specify **office**.
- Click **OK**.
- The path of the monitor attribute has now been updated with the filter condition. For example: Contract/ContractHeader/ShipToParty/Location/Address[@type="Office"]/StreetName.

Optionally, mark some attributes as "Hide in Alert". Only the attributes that are not marked as hidden are visible in the alert details. These attributes can be used in the alert message or to start a workflow from the alert.

In the alert details, the attribute label and contents for the hidden attributes are not displayed. The path to a hidden attribute and the sequence number of the parent node are still displayed in the alert data tree view.

## 6 Optionally, create conditions.

Create the conditions to use in the monitoring rule.

- In the **Conditions** tab, click **Add** to display the **Condition Builder** window.
- Specify this information:

### Name

Specify the name of the condition. For example **LateShipment** or **TargetDeliveryDateExpired**.

**Type**

Specify the condition type. Select the desired type from the list.

See [Monitor rule - condition only](#) on page 203.

**Note:** The condition types you can select depend on the attributes selected for the monitor.

- c Specify the remaining fields.

The pop-up dialog box is dynamic: the displayed fields depend on the selected condition type. For example:

- For a condition of type Attribute-Value Comparison, you must fill in these fields: **Attribute**, **Operator**, and **Value**.
- For a condition of type DateTime Check, you must specify these fields: **Time offset**, **Operator**, and **Attribute**.

**Note:**

- For details on the condition types and the information you must specify, see this section: [Monitor rule - condition only](#) on page 203
- For a condition of type Combined, you must combine the existing conditions that are using AND or OR.  
See [Defining combined conditions](#) on page 233.

- d Click **OK**. The condition is displayed in the **Conditions** tab.

- 7 Specify the monitoring rule.

Specify the monitoring rule that the monitor must use to verify incoming documents.

Complete these steps:

- a In the **Rule** tab, specify this information:

**Rule Type**

Specify the rule type. Select the desired type from the list.

See [Monitor rules](#) on page 202.

- b Specify the required information for the selected rule type. The displayed fields depend on the selected rule type.

See [Monitor rules](#) on page 202.

- c In the **Occurrence** section, specify how often an alert must be generated.

A monitor can send an alert each time the monitor rule evaluates to True. Or when the monitor rule evaluates to true several times within a specified period of time.

Complete one of these steps:

- 1 Select **Alert each time the monitor rule evaluates to true**. The monitor sends an alert each time the evaluation result of the monitor rule is true.
- 2 Select **Alert if rule evaluates to true [number] times within** and specify this information:

**Times**

Specify the number of times the monitor rule must evaluate to True, before an alert is sent.

**Within**

Specify a period of time. You must specify a number and a unit of time, such as 5 Minutes or 24 Hours. An alert is generated if the monitor rule evaluates to True the specified number of times within in this period of time. For example, you specify this information: **5 times within 24 hours**. An alert is sent if the monitor rule evaluates to true five times within 24 hours.

**Note:**

- The first time the monitor rule evaluates to true, a counter is set to 1 and a timer starts. The counter is increased each time the rule evaluates to true.
- If the counter reaches the specified number of occurrences (Times) within the specified period, an alert is generated. The counter and the timer are reset.
- If the counter does not reach the specified number of occurrences (Times) within the specified period, no alert is generated. The counter and the timer are reset.

**8** Specify the alert message.

In the **Alert Settings** tab, specify this information:

**Message**

Specify the alert message that must be displayed by the monitor. You can specify a string up to 255 characters for the Alert message or its translations. You can use document attributes that are not marked as "Hide in Alert", surrounded by square brackets, in the message text.

When you press Ctrl+Spacebar, a list of available attributes is displayed. To select an attribute, use the arrow keys and press Enter, or use the mouse. To leave the list without selecting an attribute, press Esc. If there is no value/attribute available, the Alert shows [?] in the message. For example, a monitor contains attributes SO\_ID and StandardDiscountPercent. You specify this alert message: Sales order [SO\_ID] with standard discount [StandardDiscountPercent] has lines where discount given is not within 5% of standard discount.

When an alert is generated, the attributes are substituted with the appropriate values. For example, an alert with this description is generated: Sales order SO1100359 with standard discount 3.0 has lines where discount given is not within 5% of standard discount.

**Note:** The attribute values of type string can contain up to 4000 characters. This can expand the length of the message. There is no technical limit on the number of attributes that are allowed in a message. Although, depending on the resources available on the used system, an error can happen at runtime. We recommend that you use a maximum of 10 string attributes with placeholders in one message, if all these strings are potentially long.

You can use a special ## delimiter at the end of the alert message to define a category name. The string after ## is removed from the alert message when this is displayed in Infor Ming.le. This string is listed in the **Filter** drop-down in the Alert List widget configuration and can be used to filter alerts by category. If you use a parameter placeholder after ##, the value of the parameter is used as the category name. Comparison of category names is not case-sensitive.

You can specify a different message text for several languages. Click the translation button next to the input text box. Specify a new message for each language that must be supported. To use categories, you must specify a category name in each message, for each language, using the ## delimiter. When an alert is displayed, the translation corresponding to the specified language in the user's settings in Infor Ming.le is shown to the end user. If there is no translation text for the

language selected, the message that is specified in the **Message** box is displayed. Additional language messages can be added, see [Adding translations](#) on page 328.

### Distribution

An alert is distributed to the groups or persons that are specified in the **Distribution** tab.

- a In the **Distribution** tab, click **Add**. The **Select Distribution Data** window is displayed.
- b Specify the required information.

See [Adding distribution elements](#) on page 331.

- 9 Optionally, specify drillbacks.

See [Specifying monitor drillbacks](#) on page 232.

- 10 Optionally, specify workflow settings.

Only perform this step if the user who receives the alert must be in a position to start a workflow to handle the alert.

Sometimes the user wants to start a predefined workflow to handle an alert. To enable this, link a workflow definition to the event monitor.

Before you link a workflow definition to the monitor, create the workflow definition to be used. See [Creating a workflow definition](#) on page 273.

To link a workflow definition to the monitor, perform these actions on the **Parameter Mapping** tab:

- a Select **Start Workflow from Alert**.
- b Select the workflow to be used.
- c Select the monitor attributes to be mapped to the workflow input parameters.

#### Note:

- Only workflow definitions that do not contain structures can be started from an alert. When editing the monitor, the list of available workflow definitions is filtered. You can only select workflow definitions without structures. If the configuration is changed and you click **Resolve** and try to start a workflow that has a structure, an error is reported.
- You cannot use output parameters of the workflow definition. To start a workflow that returns information to an application, use an activation policy instead of an event monitor.
- If you selected **Alert if rule evaluates to true [number] times within** in the **Rule** tab of the monitor, you cannot start a workflow from the alert. This is because the alert does not contain data to start a workflow with.

**Note:** Only attributes that are not marked as "Hide in Alert" can be mapped to workflow input parameters. For the input parameters, do not select attributes that are repeating or attributes that are part of a repeating structure in the document. For example, attributes of a sales order line.

Otherwise the starting of the workflow does not work, because the BOD can contain multiple values.

When the monitor generates an alert, and this alert is handled in Infor Ming.le a **Resolve** button is shown. Click **Resolve**, the workflow is started based on the selected workflow definition and the alert is closed. The data from the selected attributes in the alert is used as input for the workflow.

- 11 Specify distribution details.

- 12 Optionally, specify escalation details.

You can specify escalation rules. When an alert is escalated, the managers of the users from the current distribution list are informed. If the alert is unassigned during an escalation, the managers can assign it to themselves or to another user from the distribution list.

You can define several escalation levels. If no action is taken in the time interval specified in the escalation rule, the alert is escalated to the next management level.

To specify escalation details, in the **Escalations** tab, specify this information:

**Escalate when unassigned for ...**

If you select this check box, you must specify a time interval. An alert is escalated if no user has been assigned to it for the specified time interval.

**Escalate when not finished within ...**

If you select this check box, you must specify a time interval. An alert is escalated if it is not marked as Done within the specified interval of time since its creation time.

**Number of Levels**

Specify the number of management levels to which the alert must be escalated. The escalation rule is applied for each management level, as long as a user for that management level exists in Infor Federation Services (IFS).

**13** Save the monitor.

Click **Save**. The monitor definition is now complete. You can now activate the monitor.

## Specifying monitor drillbacks

Drillbacks are configured based on drillback view definition files. These files are XML files delivered by applications that must be uploaded to Infor Ming.le in the Admin Settings/Manage Drillbacks configuration.

For details, see the Infor Ming.le documentation.

For each alert, standard drillbacks are generated from the view definition files for views with level BOD associated with the document name that generated the Alert. The standard drillbacks work only based on the document ID.

To use drillbacks with several parameters or drillbacks based other types of view definitions, you can add monitor drillbacks with custom configuration. These drillbacks are generated in addition to the standard drillbacks.

To configure custom monitor drillbacks:

- 1 Select the **Drill Backs** tab in the monitor configuration.
- 2 Click **Add** to start the drillback configuration window.
- 3 Specify this information:

**Application**

Select an application name from the drop down list. These names are extracted from the drillback view definitions previously uploaded in the Infor Ming.le Admin Settings.

**View**

Select a view from the list attached with the application selected before. The View parameters are added automatically in the lower part of the window.



### View parameters

There are three standard view parameters and one or more view-specific parameters. Each parameter can be filled with a predefined value, or mapped to a monitor attribute. To specify a predefined value, select the **Value** option. To map to a monitor attribute, select the **Attribute** option.

#### Note:

If an attribute used in a drillback is a repeating attribute and it occurs several times in a document, only the first occurrence is used to generate one drillback in the alert.

If an attribute used in a drillback is not present in a document, the following happens when an alert is generated for this document:

- If the view parameter is required, this drillback is not generated.
- If the view parameter is optional, this drillback is generated and does not contain this parameter.

Parameter Type	Description
Standard parameters	<p>These are the standard parameters:</p> <ul style="list-style-type: none"> <li>• Logical ID - must be filled with the Logical ID of the application to drill to. The value must start with <code>lid://</code></li> <li>• Accounting Entity - this parameter is optional. You can specify a value of the accounting entity, or leave it blank.</li> <li>• Location - this parameter is optional. You can specify a value of the accounting entity, or leave it blank.</li> </ul>
View-specific parameters	<p>Often there is one parameter that identifies the document number, but a view can have any number of parameters. Consult the documentation of the application that delivered the Drill Back Views file about the meaning of each parameter.</p>

- 4 Click **OK** to save the new drillback configuration.
- 5 Specify the label of this drillback link in the drillbacks list view.  
Optionally, add translations. The specified label is used to show the link in the alert details.
- 6 Save the monitor.

#### Note:

- Infor Ming.le UI for Alert drillbacks shows a list with a maximum of 30 drillback links.
- You cannot define custom drillbacks for monitors that use the rule of type multi-occurrence.
- Custom alert drillbacks generated for an alert are not visible in Reports, `PulseAlert` messages, email notifications, events sent to OneView.

## Defining combined conditions

The monitor, for which you define a combined condition, must already contain at least two conditions.

- 1 Open the **Monitor** detail page for which a combined condition must be added.
- 2 On the **Conditions** tab, click **Add**. The **Condition Builder** window is displayed.
- 3 Specify this information:

**Name**

Specify the name of the condition.

**Condition Type**

Specify **Combined**. The conditions that are already linked to the monitor are displayed.

**Logical operator**

Select **AND** or **OR**.

- 4 Compose the combined condition:
  - a Select at least two conditions to combine.
  - b Click **OK**. The new combined condition is displayed on the **Conditions** tab.

**Example**

The **Conditions** tab shows the conditions A, B, and C. You want to define this combined condition: A AND (B OR C). You complete these steps:

- 1 On the **Conditions** tab, click **Add**. The **Condition Builder** window is displayed.
- 2 In the **Name** field, specify **D**.
- 3 In the **Condition Type** field, specify **Combined**. The conditions A, B, and C are displayed.
- 4 In the **Logical operator** field, select **OR**.
- 5 Select conditions B and C.
- 6 Click **OK**. The new condition, D (B OR C), is displayed on the **Conditions** tab.
- 7 On the **Conditions** tab, click **Add**. The **Condition Builder** window is displayed.
- 8 In the **Name** field, specify **E**.
- 9 In the **Condition Type** field, specify **Combined**. The conditions A, B, C, and D (B OR C) are displayed.
- 10 In the **Logical operator** field, select **AND**.
- 11 Select condition A and condition D.
- 12 Click **OK**. The new condition E (A AND D) is displayed on the **Conditions** tab. You can now use the combined condition E in the monitor rule.

## Activating monitors

You can activate new or deactivated monitors.

- 1 In the **Monitors** modeling page, select one or more monitors to activate and click **Activate**.  
The monitors start evaluating incoming application documents and sending alerts.
- 2 Create a connection point for each application, database, or message queue to monitor. Monitoring is performed on any active connection point that provides the documents that are monitored.  
To view the existing connection points and their status, select **Connect > Connection Points**.  
If no connection point is defined for the application, database, or message queue to monitor, create a connection point.

- 3 If the connection point(s) to be monitored is not yet available in an active data flow, complete these steps:
  - a Create a document flow and add a single activity for the connection point.
  - b Save and activate the document flow.

## Deactivating monitors

In the **Monitors** modeling page, select one or more monitors to deactivate and click the **Deactivate** button. The monitors stop evaluating incoming documents and sending alerts.

**Note:** All data recorded by a monitor, such as the multiple occurrence counter and elapsed waiting time, is removed. See the example.

### Example

A multi-occurrence monitor sends an alert if the monitor rule evaluates to true five times within one hour. After 45 minutes, the rule has been evaluated to true four times, so the multiple occurrence counter is 4. You deactivate the monitor. The multiple occurrence counter and the elapsed waiting time (45 minutes) are reset to 0. When you reactivate the monitor, the monitor starts again.

## Changing active monitors

You can change these settings of a monitor without deactivating the monitor:

- The alert message
- The distribution
- The escalation and reminders settings
- The drillbacks configuration

Do not deactivate the monitor if you must change one of these settings. Deactivation can cause missing alerts. When deactivating, the existing data from the monitor is archived, so you will no longer receive new alerts on that data and changes on application documents are ignored while the monitor is inactive.

In case of major changes on the monitor, such as selecting new attributes or changing the rule or conditions, you must first deactivate the monitor.

To change a monitor without deactivation:

- 1 Open the monitor detail page and make the required changes.
- 2 Save the changes and return to the list of monitors.
- 3 Activate the changed monitor to apply the changes to the active monitor in the ION Service.

## Exporting monitors

Event Management has an export/import mechanism to perform these tasks:

- Back-up and restore monitor definitions.
- Deliver monitor definitions to customers.

For example: Infor can export monitors to an XML file. Infor customers can import this file into their Event Management environment.

To export monitors:

- 1 In the **Monitors** modeling page, select the monitors to export and click **Export**.
- 2 Specify a file name for the export file and save the file.

You can now import the file into another Event Management environment.

See [Importing monitors](#) on page 236.

## Importing monitors

Event Management has an export/import mechanism to perform these tasks:

- Back-up and restore monitor definitions.
- Deliver monitor definitions to customers.

For example: Infor can export monitors to an XML file. Infor customers can import this file into their Event Management environment.

To import monitors:

- 1 In the **Monitors** modeling page, click **Import**.
- 2 Select the XML file that contains the monitors to import and click **Open**. The imported monitors are displayed in the **Monitors** page. You can now activate the monitors.

If the import file contains one or more monitors with an existing name you can choose to skip the import or to rename these monitors. If you choose to rename the monitor, it receives a postfix "\_" with a unique number.

---

## Chapter 9: Alarms

With Alarms, you can monitor documents according to business conditions in a self-service model, where end users define the monitoring rules.

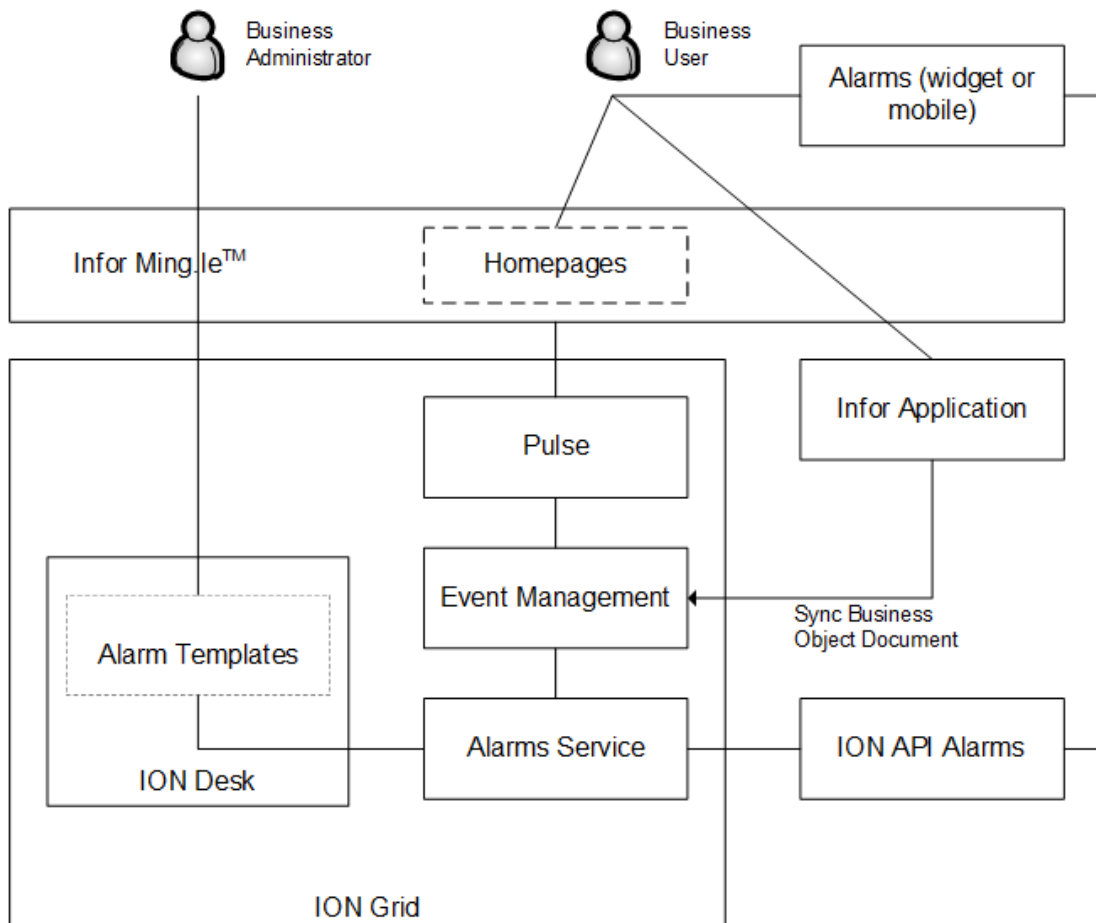
The business administrator creates alarm templates in ION Desk. You can specify role-based authorizations for each alarm template, so that only business users with the specified roles can use certain templates. Templates for which no authorizations are defined are visible to all business users after they are activated. The users create alarms based on the available templates from the Alarms mobile application or the Alarms homepage widget. Each alarm can potentially create an alert for the same document.

The alerts that are created from an alarm instance are similar to any other alerts that are created by Event Management monitors. The alarm creator can specify the distribution list for the resulting alert by selecting one or more users or one or more groups.

The users receive the alert in their personal Infor Ming.le page, the Infor Inbox application, Infor Ming.le mobile application, or by email. For details about handling alerts by end users, see the Infor Ming.le help on <https://docs.infor.com/>.

### Architecture

This diagram shows a simplified view of the Alarms architecture:



The Alarms Service and the Event Management engine are part of the ION Service installation. At runtime, the Event Management engine monitors the documents of type Sync that are configured in the alarm templates. For each template that is triggered by a document, each alarm is evaluated to determine whether an alert should be created. The communication between the Alarms server side and the Alarms widget or mobile application is done through the Alarms Service that is exposed through the ION API.

## Examples

- Alert me if payable invoice number 3456 exceeds the total amount of 10000 USD.

To configure this kind of alarm, users must select a template for the Invoice document. The template must contain the document elements for the invoice number and the total amount.

Attribute Name	Path
Invoice Number	Invoice/InvoiceHeader/DocumentID/ ID

Attribute Name	Path
Total Amount	Invoice/InvoiceHeader/TotalAmount
Currency	Invoice/InvoiceHeader/TotalAmount/ @currencyID

In the Alarms widget or mobile application, users can create two comparison conditions, which are by default joined by logical AND:

- Invoice Number = 3456
- Total Amount > 10000

Optionally, a condition can be added on the Currency code, but this filters out the documents where a different currency is used. For the alert details, users can specify a message and distribution to themselves.

## 2 Alert my team when customer ABC is added to our CRM system.

To configure this kind of alarm, users must select a template for the `CustomerPartyMaster` document. The template must contain the document element for the customer Name. For checking for the specified action "Add", the `actionCode` element is also required.

Attribute Name	Path
actionCode	(CustomerPartyMaster) DataArea/*/ActionCriteria/Action Expression/@actionCode
Name	CustomerPartyMaster/Name

In the Alarms widget or mobile application, the users can create two comparison conditions:

- actionCode = Add
- Name = ABC

For the alert details, the users can specify a message and distribution to the distribution group that represents their team.

## Concepts

### Alarm templates

Alarm templates are monitoring processes that perform these actions:

- Evaluate incoming documents.
- Extract the data corresponding to the attributes that are selected in the template configuration.
- Pass this on for evaluation of the conditions in each alarm.

The business administrator can model and activate alarm templates in ION Desk. In each template, a document is selected. From each document, a list of elements is selected to be used in conditions or to show in an alert. The business administrator also defines role based authorizations for each template. By defining and activating a template, the business administrator sets up the integration part that specifies which documents may be evaluated by alarms.

## Alarms

The business users create instances of the available alarm templates from the Alarms widget or mobile application. One alarm contains these parts:

- Comparison conditions for the document elements that the user selected from the templates
- Details about the alert to be created when the conditions are satisfied

All conditions that are defined in one alarm are joined by logical AND. One alarm creates an alert when all conditions evaluate to 'true'.

**Note:** You cannot use complex monitoring rules such as date time check, value change, document overdue, and multi-occurrence in alarms.

One user can create several alarms for each template. Each alarm has a maximum lifetime of one month. After the alarm is triggered and it created one alert or its lifetime expired, the monitoring process for this alarm stops. For more information about how to create alarms, see the *Infor ION Alarms Mobile Application User Guide* and the *Infor Ming.le User Guide*.

## Procedures

### Task overview

For alarm templates, you can perform these tasks in ION Desk:

- Configure alarm templates  
Define new alarm templates or import predefined alarm templates. See these sections:
  - [Defining alarm templates](#) on page 241
  - [Importing alarm templates](#) on page 244
- Activate alarm templates  
After activation, a template starts evaluating incoming Sync documents from the service bus. See [Activating alarm templates](#) on page 243.
- Temporarily suspend active alarm templates  
When you suspend a template, the data received as Sync documents is kept and is processed when the template is resumed. See these sections:



- 
- [Pausing active alarm templates](#) on page 405
  - [Resuming alarm templates](#) on page 406
  - Change active alarm templates  
You can change the alarm description, the template description, and the list of attributes of an active alarm template without deactivating it.  
See [Changing active alarm templates](#) on page 244.
  - Temporarily deactivate alarm templates  
In case of major changes, such as removing attributes from the template, you must deactivate the alarm template before you can change its definition.  
See these sections:
    - [Deactivating alarm templates](#) on page 243
    - [Activating alarm templates](#) on page 243
  - Deactivate alarm templates  
If an alarm template is no longer required, you can deactivate and then delete it.  
See [Deactivating alarm templates](#) on page 243
  - Export alarm templates to XML files and import alarm templates from XML files  
ION Desk has an export/import mechanism to perform these tasks:
    - Back-up and restore alarm template definitions.
    - Deliver alarm template definitions to customers.For example: Infor can export alarm templates to an XML file. Infor customers can import this file into their ION environment.  
See these sections:
    - [Exporting alarm templates](#) on page 245
    - [Importing alarm templates](#) on page 244
  - Show alarms from an active alarm template  
See [Showing alarms](#) on page 406
  - Show triggers that are processed by an active alarm template  
See these sections:
    - [Showing triggers](#) on page 406
    - [Viewing triggers](#) on page 403
  - View archived alarm templates  
See [Viewing archived alarm templates](#) on page 406.

## Modeling tasks

### Defining alarm templates

- 1 Specify generic information for the alarm template :

Specify this information::

### Name

Specify a unique name that identifies the alarm template. The name must contain the characters a-z, A-z, 0-9 and ' \_ ' (underscore). The maximum length is 255 characters. Multi-byte characters are not supported in a name.

### Alarm Description

Specify the alarm description to be displayed in the Alarms widget or mobile application. This field may contain 255 characters at most and is optional. Use this field if the template name does not contain relevant information, such as the document name that is used in this template.

### Description

Specify the description of the alarm template. This field may contain 4000 characters at most and is optional. The description is used only inside ION Desk, to describe the purpose of the monitor to the business administrator.

## 2 Select one document to be evaluated by this template

- a On the **Documents** tab in the alarm template detail page, click **Add** to display the **Select Document** window.
- b Select the document to be monitored.
- c Click **OK** to add the selected documents to the monitor.

## 3 Select the document attributes for the template.

Select several attributes that can be used by the users to create conditions or to display in alerts. After selection, you can adjust the attribute name to a more descriptive name. Only the attribute name is visible to the end user.

- a On the **Attributes** tab, click **Add**. The **Select Attributes** window is displayed.
- b Expand the tree and select the desired attributes. The selected attributes are displayed in the tooltip of the "Selected" count at the top of the window.  
For details on the attribute types you can select, see [Attribute types](#) on page 201 .
- c Click **OK**. The selected attributes are displayed on the **Attributes** tab.

## 4 Specify filters for selected attributes

Data elements and attributes that are part of a repetitive structure may require additional filtering to be used in a condition. For example, the Location from the ShipToParty information on a Contract document may exist several times. To specify which Location address details must be used in the condition evaluation, add a filter on the attribute "type" of the Location element.

Users who create alarms cannot specify filters. If a repeating element is used without a filter, only its first occurrence is used in a comparison condition.

To specify filters:

- a On the **Attributes** tab, add one or more attributes that belong to a repeating group.

For example:

```
Contract/ContractHeader/ShipToParty/Location/Address/StreetName
```

- b Select the attribute. The **Filter** button becomes enabled.
- c Click **Filter**. The **Edit Filter** window is displayed. The xml-attributes of the selected element and the xml-attributes of its repeating parent elements are displayed.
- d Specify the attribute to use for filtering. For example, specify **type**.

- e Click in the **Equals Value** column and specify a value. For example, specify **Office**.
- f Click **OK**.
- g The path of the monitor attribute has now been updated with the filter condition.

For example:

```
Contract/ContractHeader/ShipToParty/Location/Address[@type="Office"]/StreetName
```

- 5 Optionally, specify authorizations for the alarm template.
  - a Select the **Authorizations** tab.
  - b Click the plus icon to add a security role. In the **Add Authorization** dialog box you can filter on the role name and description. Select the applicable security role and click **OK** to return to the authorizations list. Users with this role are allowed to create alarms using this template. Security roles are created in the Infor Ming.le User Management application.
  - c Add or delete roles as required.
 

If no security roles are added to the **Authorizations** tab, this template is available in the Alarms widget to all Infor Ming.le Homepages users.
- 6 After the definition of an alarm template is finished, click **Save**. You can now activate or export this definition.

## Activating alarm templates

To activate a new alarm template or an alarm template that was previously deactivated:

- 1 Open the alarm templates modeling page.
- 2 Move the pointer over the template tile.
- 3 Click **Activate**.

Alternatively, click **Activate** from the detail view of a template.

After activation, an alarm template starts monitoring documents and is visible to the users in the Alarms widget or mobile application. Users can now create alarms for this template.

To ensure that documents are routed to this template, you must also configure connection points and documents that publish the correct type of documents.

See [Activating monitors](#) on page 234.

To re-activate an active template, click **Activate**. This is required in these situations:

- The alarm template was changed.  
See [Changing active alarm templates](#) on page 244.
- The document definition was changed. For example, a code list definition was extended.
- The authorizations for the alarm template were changed.

## Deactivating alarm templates

- 1 In the **Alarm Templates** modeling page, select one or more templates to deactivate.

## 2 Click **Deactivate**.

The templates stop evaluating documents and they are not visible anymore to end users to create alarms. If there were alarms that have not yet completed at the time of deactivation, these are forcefully stopped. In this case the alarms show status `Canceled` in the mobile application or widget.

## Changing active alarm templates

You can change these settings of an alarm template without deactivating it:

- The alarm description that is displayed in the Alarms widget or mobile application.
- The description of the template.
- Add more attributes to the alarm template.
- Add or remove security roles from the list of authorizations for this template.

Do not deactivate the alarm template if you only must change one of these settings. Deactivation can cause alarms to be forcefully stopped and miss alerts. When deactivating, application documents are ignored when the template is inactive. In case of major changes on the alarm template, such as removing attributes, you must first deactivate it.

**Note:** Existing active alarms go in `Canceled` state when you deactivate the related alarm template.

To change an alarm template without deactivation:

- 1 Open the alarm template's detail page and make the required changes.
- 2 Save the changes and return to the list of templates.
- 3 Activate the changed template to apply the changes to the active alarm template in the ION Service.  
The changes are visible to the users of the Alarms widget or mobile application the next time they create an alarm for this template.

**Note:** If you clone or re-use an alarm that was created with a previous version of an alert template. The alert does not contain the latest alert template changes.

After authorizations for a template are removed, any running alarms can finish. You cannot clone alarms or create new alarms for a template for which authorizations are restricted.

## Importing alarm templates

- 1 In the **Alarm Templates** modeling page, click **Import**.
- 2 Select the XML file that contains the alarm templates to import. Click **Open**. The imported templates are displayed in the **Alarm Templates** page. You can now activate the templates.

If the import file contains alarm templates with an existing name, you can choose to skip the import or to rename these templates. If you choose to rename the template, it receives a "\_" postfix with a unique number.

**Note:** If an alarm template has authorizations configured for a role name that does not exist, import and activation succeeds without warning. The security roles can be imported independent from the alarm templates import.

## Exporting alarm templates

- 1 In the **Alarm Templates** modeling page, select one or more templates.
- 2 Click **Export**.
- 3 Specify a file name for the export file and save the file.

You can now import the file into another ION environment. You can also import the file back into the same ION environment, for example to create backups or clones of the templates.

See [Importing alarm templates](#) on page 244.

---

## Chapter 10: Workflow

Workflow enables communication from an application to a user.

For example, it presents a document for approval or prompts the user to perform a task using a particular screen of an application.

A workflow is a sequence of steps that completes a business goal. These are examples of workflows:

- An approval flow which can be a sequence of approval requests to several approvers
- A review flow which can be a set of parallel tasks that is sent to multiple users to review the same document

The workflow steps can be performed by various users, based on the distribution rules for each step.

Workflow in ION provides functionality to define a workflow using a graphical modeler to create a diagram in a BPMN notation. The workflow definitions can immediately be activated in the workflow engine.

Tasks that are generated by Workflow in ION are displayed in the user's Infor Ming.le page, the Infor Inbox application and the **Task List** widget. Workflow execution waits for user action before continuing execution. You can view the workflow diagram in **Workflow Viewer widget**, with a graphical indication of the workflow progress.

### Working with a set of workflow definitions

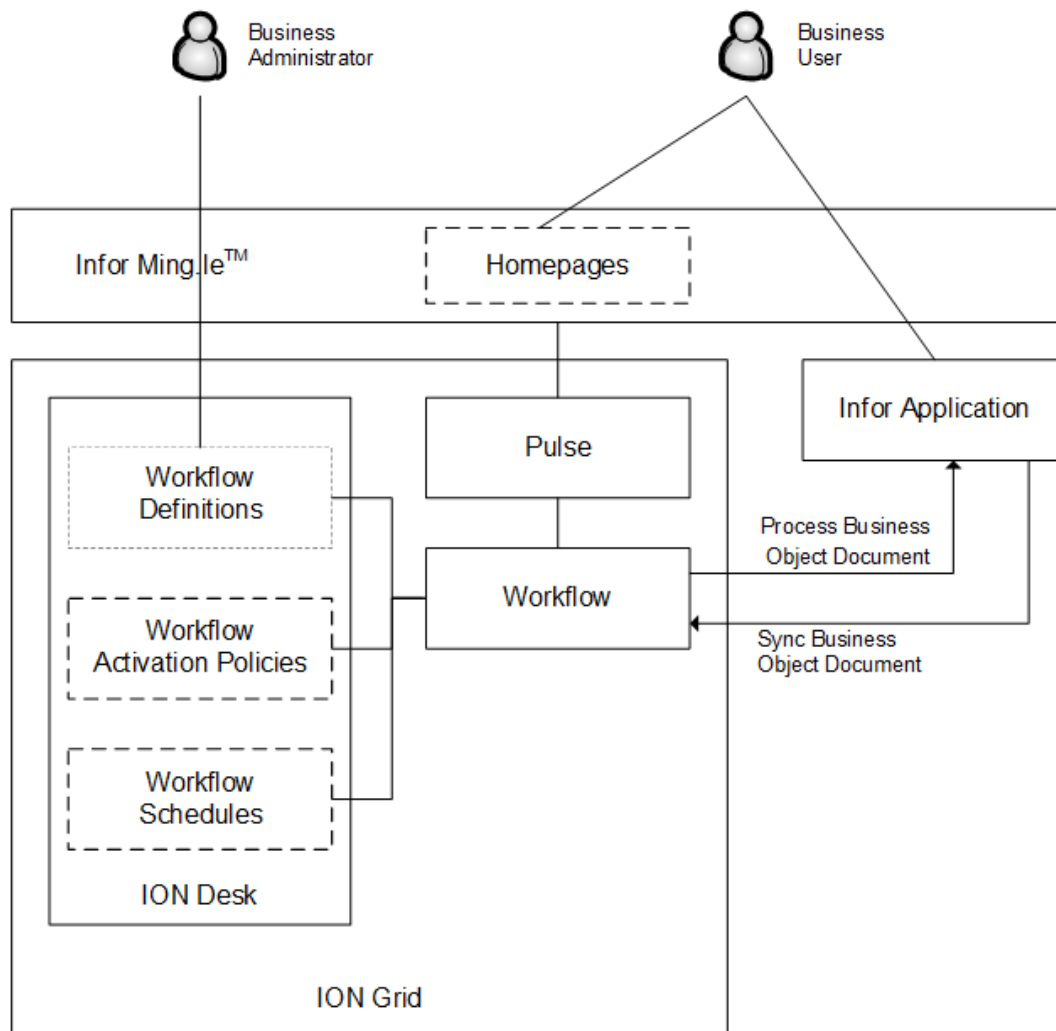
Workflow definitions can be created, edited, duplicated, and deleted in the modeling environment. As long as the definitions are not activated in the ION Service, any changes can be done.

When a workflow definition is completed, it can be activated. This means the workflow definition is deployed to the ION Service and is available for execution.

A set of workflow definitions can be exported to a file, which can be imported in another installation of ION.

## Architecture

This diagram shows the overall architecture of Workflow in ION:



## Starting a workflow

After a workflow definition is activated, the workflow can be started. Whenever a workflow is started, a new workflow instance is generated. If a workflow is started multiple times, multiple instances of that workflow exist.

This table shows methods to start workflows:

Starting method	Description
Activation policies	<p>Workflow instances can be started through Workflow Activation Policies. Activation policies can be compared with event monitors; they react on Sync documents and trigger workflow execution based on rules on data from these documents. The workflow response is sent back as a Process document message.</p> <p>See <a href="#">Activation policies</a> on page 318.</p>
Workflow schedules	<p>You can create a workflow schedule to start a workflow instance at regular intervals of time. You can specify constant values for the workflow input parameters. The workflow output parameters are ignored. Workflows that contain structures cannot be started by a workflow schedule.</p> <p>See <a href="#">Workflow schedules</a> on page 323.</p>
Workflow activity in document flow	<p>A workflow can be linked to an activity in a document flow. In that case, when a document arrives at this activity, the workflow is started. Data from the document is used as input for the workflow. When the workflow is completed, the output of the workflow is added to the document, and the document flow continues. If there is no next activity in the flow then the workflow output is ignored.</p>
Process workflow	<p>Workflow instances can be started by sending a Process Workflow Document directly to the Workflow component in ION.</p>
Start Workflow from Alert	<p>A workflow can be started by a user from an alert the user received. The workflow definition to be started and the workflow input parameters are defined in the monitor that generated the alert.</p>
Start Workflow activity in a workflow	<p>You can start a workflow from another workflow. You can start the workflow synchronously and wait for its results, or asynchronously and continue with the current flow.</p> <p>See <a href="#">Specifying details for a start workflow element</a> on page 305.</p>
Start Workflows homepage widget	<p>A workflow can be started manually by a business user who has the required workflow authorizations.</p> <p>For details about starting a workflow manually, see <a href="#">Workflow authorizations</a> on page 313 and the <i>Infor Ming.le User Guide</i>.</p>
Start workflow from API	<p>From version 12.0.34, the <code>/workflow/start</code> API method has been added to the endpoint <code>process/application</code> in the ION Services suite that is exposed in ION API. For technical details, see the endpoint swagger documentation.</p>

Activation policies and workflow activities in a document flow are comparable. The workflow is triggered by any type of document and any type of connection point.

Any type of documents are:

- ExpenseReport
- SalesOrder
- PlanningSchedule



Any type of connection points are:

- Application
- Database
- Web service

In both activation methods, also attributes from the document are mapped to the input parameters of the workflow. The output parameters of the workflow are mapped to attributes of the document again.

This table shows the differences between these two activation methods:

Workflow Activity	Activation Policy
Modeled in the context of a document flow. Only matching BODs from the preceding step in the flow are used in the workflow activity.	The routing of BODs to the activation policy is not explicitly modeled. A matching document that enters ION from any connection point triggers the activation policy.
Unconditional. Note that filters or content-based routing can be used in the document flow.	The specified rule and conditions are evaluated.
When the workflow is completed, the message that triggered the workflow is enriched with the outcome of the workflow and sent to the next activity in the flow.	When the workflow is completed, to communicate the outcome of the workflow, a Process message is sent back to the connection point that published the Sync message.
The workflow can only be canceled by an administrator through ION Desk.	An additional activation policy can be used to automatically cancel the workflow if it is no longer relevant.

Multiple ways of starting a workflow can be combined for a single workflow definition. For example, a workflow that has an activation policy can also be started using a `ProcessWorkflow` BOD.

**Note:** We recommend that you import and export workflow definitions and related items, such as activation policies or document flows, together.

## Workflow definition

A workflow definition can contain these predefined types of step:

- Task
- Task Chain
- Notification
- Decision Table
- Set Parameter
- ION API
- Wait
- Start Workflow
- Decision

- Loop Back
- Parallel
- Subprocess

See [Workflow elements](#) on page 252.

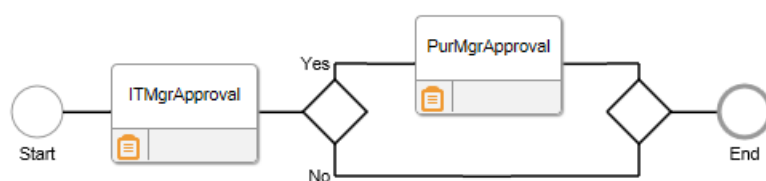
All steps can be combined without restrictions. A workflow definition can contain multiple steps of the same type.

A workflow definition also contains these parts:

- The parameters that are used in the workflow steps.  
See [Workflow parameters](#) on page 255.
- The conditions that are used in the workflow steps.  
See [Workflow conditions](#) on page 257.
- The drillbacks that are used on workflow tasks and notifications.  
See [Workflow drillbacks](#) on page 261.
- The structures that are used as input data and used in workflow steps.  
See [Workflow structures](#) on page 264.

## Sample workflow

This diagram shows a sample workflow to approve purchase requisitions:



This table shows the elements in the workflow:

Type	Name	Description
Task	ITMgrApproval	<p>This Task:</p> <ul style="list-style-type: none"> <li>Is distributed to multiple users: <ul style="list-style-type: none"> <li>All the users in the distribution list see this Task, but only one user can approve the Task.</li> <li>To let several users or roles approve in a chain, multiple Tasks should be created.</li> </ul> </li> <li>Asks these users to approve a requisition.</li> <li>Has various parameters, such as DocumentID, RequesterName, TotalAmount.</li> </ul> <p>These custom action buttons are defined:</p> <ul style="list-style-type: none"> <li><b>Approve</b></li> <li><b>Reject</b></li> </ul>
Decision	N/a	<p>A condition is linked to this element. This condition checks whether the requisition was approved in the previous step and the amount is high enough to ask the director to approve the requisition: ApproveResult=Approve and Amount &gt; \$ 1,000</p> <p>If the condition is true, that is, if the requisition was approved and the amount is more than \$1,000, the workflow follows the Yes branch.</p> <p>If the condition is false, that is, if the requisition was rejected or the amount is less than or equal to \$ 1,000, the workflow follows the No branch.</p>
Task	PurMgrApproval	<p>This Task is distributed to the PurchasingManager group. The PurchasingManager is asked to approve the requisition.</p>
Parameter	DocumentID, TotalAmount, ApproveResult, RequesterName	<p>These parameters are used in the Decision and Task elements in the workflow. The ApproveResult is output for this workflow.</p>

## Concepts

### Workflow patterns

These workflow patterns are supported:

- Sequential flow with activities of type Task, Notification, and so on.
- Conditional flow using a step of type Decision

A conditional flow has two branches, "Yes" and "No", and uses a condition based on the workflow parameters.



- Repeating flows using a Loop Back step
- Parallel flows using a Parallel step
  - An unlimited number of branches is allowed and parallel execution is unconditional.
  - You can configure whether the workflow must continue when all branches are completed or when one of the branches is completed.








- Subprocess




This is a modeling optimization to delimit a part of the workflow that can be expanded and collapsed. A subprocess is not a workflow on itself.

## Workflow elements

This table shows the elements that can be used in a workflow definition:

Step	Description
<p>Task</p> 	<p>A workflow step that creates an entry of type Task in a user's Task list. The workflow suspends execution until a user has completed this Task, and then moves to the next step in the flow.</p> <p>Tasks can be configured with these features:</p> <ul style="list-style-type: none"> <li>• Display Workflow Parameter The user can view the current value of a workflow parameter of any type.</li> <li>• Display and edit Workflow Parameter The user can view the current value of a workflow Parameter and specify/enter a new value.</li> <li>• Specific action buttons The default action button for a Task is <b>Done</b>. This button notifies the workflow that the Task is completed. You can also configure multiple 'special' buttons that also notify the workflow but also set a (String) parameter to a specific value.</li> <li>• Display the task details in a custom form By default, the task details are displayed in the Task List widget. You can also configure a link to another application that hosts the form to perform a task. The user must close the task using the external user interface.</li> </ul> <p>Each Task is distributed to one or more users as specified in the Distribution list. All users in the distribution list see when a Task is created, but only one user at the time can work on a Task.</p> <p>If the distribution type of a Task is specified as Parallel, all users in the distribution list receive a copy of this Task. The Task is completed only when all the users from the distribution list have marked their copies as Done.</p> <p>For each task, escalation and reminders can be defined. Users can receive an email reminder before a task is escalated. In case of an escalation, users or their managers are added to the distribution list so that they can take actions on this task.</p>
<p>Task Chain</p> 	<p>A series of approval tasks that are sent to different people. An approval matrix is used to model who should approve. If the user who picks up the first task approves, a new task is created if more people must approve. If a user rejects, the task chain stops.</p>

Step	Description
Notification 	<p>A workflow step that creates an entry of type Notification in a user's Task list. The workflow continues execution after the Notification is sent to a user. Notifications can be configured with these features:</p> <ul style="list-style-type: none"> <li>• Send a message to inform the user that a certain point in the workflow is reached.</li> <li>• Optionally, include notes from previous steps.</li> <li>• Optionally, specify an expiry time.</li> <li>• Display workflow parameters and structures.</li> </ul> <p>The user can view the current value of workflow parameters.</p> <p>A distribution list can be defined for Notifications in the same way as for the Tasks. Notifications are always distributed individually to each user in the distribution list.</p>
Set Parameter 	<p>An automatic step to change the values of parameters. Set Parameter elements are used to perform these tasks:</p> <ul style="list-style-type: none"> <li>• Assign a value to a parameter.</li> <li>• Assign the value of a parameter to another parameter.</li> <li>• Assign an expression to a parameter.</li> <li>• Assign the Common Name of a user from IFS to a parameter.</li> <li>• Assign a value from a structure to a parameter.</li> </ul>
Decision Table 	<p>An automatic step to set the value for one or more parameters that are based on the evaluation of a complex set of conditions. Several conditions can be defined as table rows. The evaluation of the conditions follows the order in the table from top to bottom. The evaluation stops when the first combination of conditions that are True is encountered.</p>
ION API 	<p>A step to perform a call to an ION API operation. In this step, you can use workflow input parameters, structures, and task notes for the operation input. You can extract values from the output response and map them into workflow parameters.</p>
Wait 	<p>A step to let the workflow execution wait for a specified amount of time or until a specified date and time.</p>
Start Workflow 	<p>A step to start another workflow. You can model to start a workflow synchronously and wait for its results, or to start asynchronously and continue the current workflow.</p>
Decision 	<p>An evaluation of a condition that has two possible follow up branches: Yes and No. If the evaluation result is true, the workflow continues execution following the Yes branch. All the steps from the No branch are not performed.</p> <p>If the evaluation result is false, the workflow continues execution following the No branch. All the steps from the Yes branch are not performed.</p> <p>Conditions are created using comparison operators and the workflow parameters. To create combined conditions, use the logical operators AND and OR.</p> <p>See <a href="#">Workflow conditions</a> on page 257.</p>

Step	Description
Loop Back 	<p>A part of the flow that is repeated. When the last activity inside the loop back is completed, the workflow either continues or recycles at the start of the loop back. You must configure a maximum number of loops and a condition. The workflow recycles if the condition is met and the maximum number of loops is not reached yet. The workflow does not recycle if the condition is not met or the maximum number of loops is reached.</p>
Parallel 	<p>An unconditional execution of two or more execution branches in parallel. When the workflow execution reaches a parallel split, all the branches are started simultaneously. There is no guarantee in which order the branches are completed. You can configure when the workflow must continue:</p> <ul style="list-style-type: none"> <li>• When all branches are completed. In this case the workflow waits until the activities in each of the branches are completed.</li> <li>• When one branch is completed. In this case the other branches are canceled as soon as all activities in one of the branches are completed.</li> </ul>
Subprocess 	<p>A part of a workflow model that can be expanded or collapsed in the modeler. Use subprocesses if you are modeling a large workflow that can be split up in multiple groups of items that belong together; this makes modeling easier. For example, a workflow can consist of two parts: engineering and planning. Each of these subprocesses can contain multiple Tasks, Notifications, Decisions, and so on. When working on the engineering part, you can collapse the planning subprocess. A subprocess can contain other subprocesses. Using subprocesses does not change the behavior of the workflow in the ION Service.</p>

## Workflow parameters

Workflow parameters contain data that can be used in various steps of the workflow. Every step of the workflow has access to the complete set of the workflow parameters.

The workflow starts with the complete set of these parameters. The values of the parameters are specified through one of these means:

- Input parameters that are passed to the workflow at the moment it starts
  - Initial value, specified in the workflow definition
- Every parameter that is not an input parameter, must have an initial value.

When the workflow reaches the End step, every parameter will have a value. The values for the output parameters are reported back to the initiator of the workflow.

A parameter can be an input parameter, an output parameter, both input and output, or neither input or output.

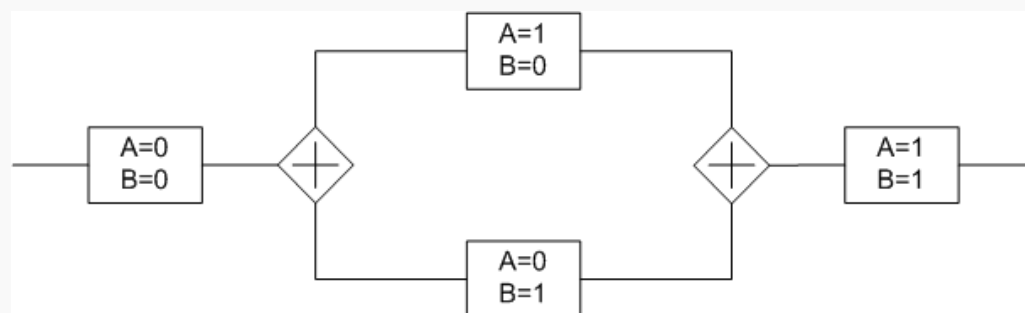
## Parameters in parallel execution

Each serial flow in the workflow has a set of parameters. Therefore, if flows run in parallel, multiple parameter sets exist simultaneously. Where parallel flows are combined with a merge step of type all-in, these sets must be merged.

When the parameter sets are merged, and a parameter is changed in one of the branches of a parallel flow, the changed value prevails over the unchanged values in the other branches.

### Example

This diagram shows an example:



Parameter A is changed in the upper branch of a parallel flow. Parameter B is changed in the lower branch. After merging the branches, both parameters keep the changed value (1).

**Note:** If the same parameter is changed in two parallel branches, the resulting value after merging those branches is undetermined. It can be either value.

## Parameter data types

This table shows the data types:

Data type	Description
String	A string with maximum length of 4000 characters.
Integer	A (64 bits signed) whole number
Decimal	Numbers that include a decimal point. The Decimal data type in Workflow corresponds to the double data type and is a double-precision 64-bit IEEE 754 floating point.  Such numbers are shown with a maximum of 15 significant decimal digits in the user facing UI such as homepage widgets and email notifications. In ION Desk, the actual values of the decimal numbers are shown and can be represented in exponential notation.
Boolean	True or False
Date	Contains a date value



Data type	Description
Date And Time	Contains a date and time value
Code	A string having a value selected from a Code
Hyperlink	A string containing a hyperlink starting with <code>http://</code> , <code>https://</code> or <code>ftp://</code> .
User	A string containing the personid of an IFS user
DistributionGroup	A string containing an IFS distribution group

## Workflow conditions

A workflow definition can use comparison conditions that are based on parameters. You can also define combined conditions.

Workflow in ION supports these condition types:

- Parameter comparison
- Parameter-value comparison
- Combined condition
- 'Any Value', used in decision tables and approval matrices

In conditions, you can use various comparison operators.

See [Operators in workflow conditions](#) on page 260.

## Parameter-value comparison

To compare the value of a parameter to a fixed value, use a parameter-value comparison.

This table shows the syntax of a parameter-value comparison:

<b>Syntax</b>	[Parameter][Operator][Value]
<b>Parameter</b>	One of the parameters that are defined in the workflow definition.
<b>Operator</b>	Comparison operator; depends on the parameter type. See <a href="#">Operators in workflow conditions</a> on page 260.
<b>Value</b>	A constant value that is of the same type as the parameter.
<b>Evaluation</b>	The evaluation result is True or False.

### Example

This table shows an example of a parameter-value comparison:

<b>Parameter</b>	StartDate
------------------	-----------

<b>Operator</b>	>
<b>Value</b>	2005-01-01

The evaluation result is true if the Date value of the StartDate is after January 1st 2005.

## Parameter comparison

To compare the values of two parameters that are selected in a workflow definition, use a parameter comparison. Both parameters must have the same data type.

This table shows the syntax of a parameter comparison:

<b>Syntax</b>	[Parameter1][Operator][Parameter2]
<b>Parameter 1</b>	One of the parameters that are defined in the workflow definition. The parameter can be of these types: <ul style="list-style-type: none"> <li>• String</li> <li>• Integer</li> <li>• Decimal</li> <li>• Boolean</li> <li>• Date</li> <li>• Date And Time</li> <li>• Code</li> <li>• User</li> <li>• DistributionGroup</li> </ul>
<b>Operator</b>	Comparison operator; depends on the parameter type. See <a href="#">Operators in workflow conditions</a> on page 260.
<b>Parameter 2</b>	One of the parameters that are defined in the workflow definition. The parameter is of the same type as Parameter 1.
<b>Evaluation</b>	The evaluation result is True or False.

### Example

This table shows an example of a parameter comparison:

<b>Parameter 1</b>	OrderedQuantity
<b>Operator</b>	<
<b>Parameter 2</b>	ShippedQuantity

The evaluation result is True when the comparison between the values of the two parameters is True.

## Combined condition

To evaluate several condition types simultaneously, create a combined condition. To combine all condition types, use the logical operators AND and OR.

The evaluation result of a combined condition is True or False, based on the evaluation results of the sub-conditions.

This table shows the syntax of a combined condition:

<b>Syntax</b>	[Condition1] AND   OR [Condition2] <b>Note:</b> You can have more than two sub conditions in one combined condition. For example: [Condition1] AND [Condition2] AND [Condition3]
<b>Condition1</b>	A condition that is already defined in the workflow definition.
<b>Condition2</b>	A condition that is already defined in the workflow definition.
<b>Evaluation if AND is used</b>	The evaluation result is True if all conditions are True.
<b>Evaluation if OR is used</b>	The evaluation result is True if at least one of the conditions is True.

This table shows the syntax of a nested combined condition:

<b>Syntax</b>	[Condition1] AND   OR ([Combined Condition])
<b>Condition1</b>	A condition that is already defined in the workflow definition.
<b>Combined Condition</b>	A combined condition that is already built using AND or OR, based on the list of conditions. Combined conditions that are reused are surrounded by round parentheses.
<b>Evaluation</b>	The comparison conditions are evaluated first. The combined conditions that are marked by parentheses are then evaluated. The evaluation result is True or False.

### Example

This table shows an example of a combined condition:

Name	Type	Condition	Description
SalesOrderShipped	Parameter-Value Comparison	SOStatus = Shipped	Checks whether a sales order is shipped.
PartialShipment	Parameter Comparison	ShippedQuantity < OrderedQuantity	Checks whether only part of the order was shipped.

Name	Type	Condition	Description
DelayedShipment	Parameter Comparison	PromisedShipDate-Time < ActualShipdate-Time	Checks whether the shipment was late.
PartialOrDelayed	Combined Condition	PartialShipment OR DelayedShipment	Checks whether the shipment is partial or delayed.
PartialOrDelayedShipment	Combined Condition	SalesOrderShipped AND PartialOrDelayed	Checks whether a partial or delayed shipment took place.

The resulting condition is:

SOSStatus = Shipped AND (ShippedQuantity < OrderedQuantity OR PromisedShipDateTime < ActualShipdateTime)

## Operators in workflow conditions

This table shows the operators that you can use in conditions:

Parameter Data Type	Operator	Description
String or Code	=	True if strings match (case sensitive)
	<>	True if strings do not match (case sensitive)
	starts with	True if the attribute starts with the specified string value
	contains	True if the attribute contains the specified string value
	ends with	True if the attribute ends with the specified string value
	in set	True if the attribute is equal to one of the values specified in the Set. Set is a list of string values separated by commas.
	in codes	True if the attribute is equal to one of the codes selected from a specified code

Parameter Data Type	Operator	Description
Integer, Decimal, Date, or Date And Time	<	Less than
	<=	Less than or equal to
	>	Greater than
	>=	Greater than or equal to
	=	Equals
	<>	Does not equal
	between	True if the value of the attribute is between two specified values By specifying two identical values, you can use this operator to compare a date to a specific day/hour/minute.
Boolean, User, Distribution-Group	=	Equals
	<>	Does not equal

## Workflow drillbacks

You can use a Workflow drillback to create an Infor Ming.le drillback on a task or notification. The drillback is based on a view definition and contains a mapping from workflow parameters towards view parameters.

Like parameters, configured Workflow drillbacks can be used on the task or notification **Content** tab to model the contents of the notification or task.

In task and task chain steps, a drillback can also be used to specify that the task should be executed from an external custom form. Such a form is hosted outside of the Task List widget.

Workflow drillbacks are configured based on drillback view definition files. These are XML files delivered by applications that must be uploaded to Infor Ming.le in the Admin Settings/Manage Drillbacks configuration. For details, see the Infor Ming.le documentation.

## Workflow tasks

A workflow can contain a task step representing an activity that must be performed by a user. The advantage of using the Workflow is that you can model how these tasks are performed and that the results of the user actions can be used in subsequent workflow steps.

For all features of a task step, see [Specifying details for a task element](#) on page 280. In this section, we describe advanced functionality related to workflow tasks.

## Canceling workflow tasks

Workflow tasks that are generated by a workflow model can be canceled by an administrator by canceling the workflow instance that generated this task.

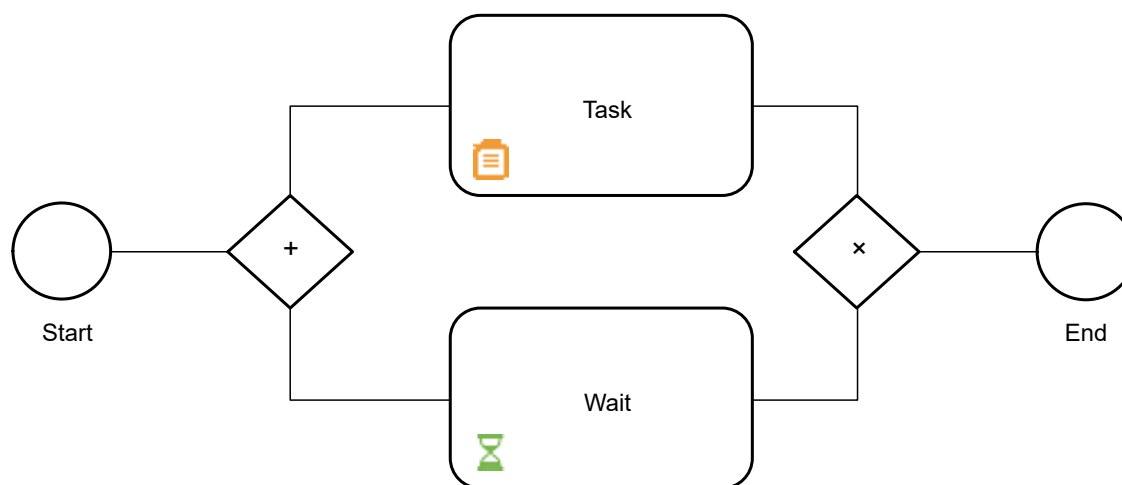
- 1 In ION Desk, select **Monitors & Workflows > Activities**.
- 2 Search for the task to be canceled.  
You can search on the task Id.
- 3 Select the task row and click **SOURCE** to see the workflow instance that generated this task.
- 4 Select the workflow instance row and click **CANCEL**.  
When the workflow instance is canceled, all outstanding tasks created by this workflow are canceled.  
See [Canceling a workflow instance in ION Desk](#) on page 317.

## Configuring automatic task cancellation

Configure a workflow model to automatically cancel a task that was not completed within a given time interval after the task was created. You must use a workflow step of type parallel.

- 1 Add a workflow step of type parallel flow with two branches.  
In the step properties, select the **One branch completed** option.  
See [Specifying details for a parallel element](#) on page 307
- 2 Place the workflow task on one of the branches.  
See [Specifying details for a task element](#) on page 280.
- 3 Place a workflow wait step on the other branch.  
See [Specifying details for a wait element](#) on page 304.

This is how the workflow looks like:



At runtime, both the task activity and the wait step are triggered simultaneously. If the task is not completed within the amount of time specified in the wait step, the branch with the wait step

completes first. Then the workflow continues to the next step. The task is canceled, and any workflow parameters used in this task remain unchanged.

If notes have been added to the task that was canceled, these notes are not visible in other workflow tasks after the parallel step.

If attachments have been added to the task that was canceled, they are visible in other workflow tasks after the parallel step.

## Workflow tasks with custom forms

You can redirect the execution of a task to a custom form that is hosted by another application that runs inside Infor Ming.le. In this case, the users receive the tasks in the **Task List** widget or the Infor Inbox application. When they choose to view the details, they are redirected to the form that is hosted in another application.

The application that hosts the form can use the APIs that are exposed in the endpoint process/user in the Infor ION suite in ION API. This API provides methods to show the task details on the form and to close the task from the form.

For more details on the available APIs, see the *Infor ION Development Guide* and the endpoint Swagger documentation.

## Using custom forms to perform workflow tasks

To configure that a workflow task may only be executed from an external application form:

- 1 In your application, implement a screen that hosts the task form. To retrieve the task details based on a task ID, you can use the exposed ION Process APIs.

You must ensure that this application form has these features:

- It can retrieve the IFS Person ID of the user currently logged in from Infor User Management, also known as IFS.
- It can call the ION Process APIs from endpoint process/user in the ION services suite in ION API.
- The task actions are represented by buttons on the form as configured in the workflow model. When the user clicks one of these buttons, the task is closed with the correct value for the selected action.

Example:

- A Task is configured with two action buttons: **Approve** with value 1 and **Reject** with value 0.
- These buttons must be displayed on the form. The button default labels or translated labels can be displayed as the form's buttons.
- For example, if the user clicks **Approve**, the value 1 must be used in the API call to close the task.
- The form is accessible through an Infor Ming.le drillback and it can receive at least one parameter containing the task ID. This ID is required to retrieve the details of the correct task to show on the form.

- 2 Create a view definition file for the application that contains the form. In this view definition file, specify the view definition for the drillback to the form. Upload this file in Infor Ming.le in the Manage Drillbacks section from the Administration Settings.

For more details about managing applications and their drillback files, see the Infor Ming.le online help.

Example of a view definition for a form:

```
<View name="LinkToForm" level="OTHER" standard="no">
  <description>Link to open a task form</description>
  <parameter name="ID1" required="yes">FormId</parameter>
  <parameter name="ID2" required="no">Theme</parameter>
  <parameter name="ID3" required="yes">TaskId</parameter>
</View>
```

- 3 In the Workflow Modeler, create a drill back and select the view that is defined for the form. In the **Parameters** column, map the view input parameter that is meant to contain the task ID to the <ACTIVITY\_ID> placeholder.
- 4 Create a workflow step of type Task or Task Chain with all properties. Create this step in the same way, as you would do for a normal user task. On the **Content** tab, select the **Use Custom Form** check box and select the drill back to the form. At runtime, the placeholders in this link are replaced with the actual values. The parameter that is mapped to the <ACTIVITY\_ID> placeholder is replaced with the id of the current task.

**Note:**

- If a custom form is used for a Task chain, the same form is presented for each approval task.
- An error is displayed in these situations:
  - The link to the application form is not correct.
  - The application that hosts the form is not running when a user opens the details of this task in the Task List widget.

The error message contains the task ID which can be used for troubleshooting. Alternatively, the user can move the pointer over the task icon to see the task ID.

The system administrator can look up this task in the **Manage Activities** page in ION Desk and view its details. If required, the system administrator can also find the workflow instance for this task and cancel it.

- You cannot use custom forms for Alerts and Notifications.
- The **Use Custom Form** flag and the URL link are not included in the `Sync.PulseTask` BOD.

## Workflow structures

Similar to workflow parameters, workflow structures are definitions of complex data structures that can be used to pass input data into the workflow. The workflow structures cannot be used as output. A workflow structure is by default required input, so when triggering the workflow, values must be passed over for structures.



A workflow structure is a collection of levels and fields. Each level can have one or more children that are fields or sublevels. There is no limit in the degree of nesting or the number of children allowed for each level.

**Note:** When triggering a workflow, a workflow structure with at most 2000 levels and fields can be passed as workflow input.

The fields in a structure are values that can have these data-types:

- Boolean
- Decimal
- Integer
- String
- Date
- Date and Time

Each level in a structure may be repeating an indefinite number of times; each field is single occurrence.

When starting a workflow through workflow activation policies or from document flows, use structures to map data from business documents while preserving the nested structure. When a workflow is started through a Process.Workflow BOD, structures are used to pass complex input data to workflow. This data is displayed as a nested structure in the workflow Tasks and Notifications.

Workflows that contain structures cannot be started on demand from Infor Ming.le. They also cannot be started from alerts.

In a workflow definition, structures can be used in these types of steps:

- Tasks and Notifications, including the e-mail notifications
- Tasks generated by steps of type Task Chain and Parallel Task
- Workflow exit (input only)
- Set Parameter (read only)

Workflow structures can be used in the details of Tasks and Notifications and can be sent as input to the workflow exits. With a step of type Set Parameter, data can be extracted from the structures and assigned into workflow parameters for further calculations or decisions. For details, see the "Procedures" section about specifying details for each kind of step.

## Workflow ION API call

The functionality to call an ION API is available in Workflow and contains this functionality:

- You can use a new workflow step named "ION API" to model an API call at any place in the workflow.
- You can call an ION API operation that is exposed through the ION API metadata. Reply of type JSON, XML, and Text is accepted
- You can use both, reading and writing operations.
- You can search for the operation to use based on the ION API metadata. The ION API Gateway applies an elastic search algorithm.
- You can configure the security credentials for each workflow ION API step by importing a service account.

- You can map between API request and API output parameters and workflow parameters.
- You can pass workflow structures and task notes as request body for POST and PUT methods, which support these data structures.
- You can test the API execution during modeling.
- You can configure preferences for error handling.

These features are not supported:

- Reply of REST Streaming Methods
- Input parameters of type FORMDATA and input content type "multipart/form-data". Only parameters of type FORMDATA and input content type "application/x-www-form-urlencoded" are supported. If the input content type is not specified, it is assumed the type is "application/x-www-form-urlencoded".

To model a workflow that makes an ION API call, start ION Desk and navigate to the workflow modeler. Add a workflow step of type ION API and specify its properties as described later. When the configuration is ready, save and activate the workflow.

See [Specifying details for an ION API element](#) on page 299.

To call an API, these configuration steps must be completed:

- Ensure the application that exposes the ION API to call is registered and its endpoints are indexed. See "Registering the client application" in the *Infor ION API Administration Guide*.
- In the Infor Ming.le User Management UI, create service accounts for the backend application that exposes the ION APIs to call. Ensure to specify the description property of the service account, as this is required by ION. See "Service Accounts" in the *Infor Ming.le Cloud Edition Administration Guide*.

## References

The workflow ION API step is built using the ION API metadata exposed by the ION API Gateway. Information about the ION API functionality can be found in these documents:

- *Infor Ming.le Cloud Edition Administration Guide*
- *Infor ION API Administration Guide*

## Specifying a path for output mapping

In output parameters mapping you must specify a path for a mapping with response of type BODY.

If the response content type is application/json, you must specify a JSON Path. For more information on JSON Path syntax, see the Internet.

If the response content type is application/xml, you must specify an XPath. For a reference of the XPath syntax, see: [http://www.w3schools.com/xml/xpath\\_syntax.asp](http://www.w3schools.com/xml/xpath_syntax.asp).

When using a simple path that points to an element that is repeating, or is part of a structure that is repeating, the first occurrence is retrieved by default. To extract a different occurrence, you can use index and filter expressions. See the examples.

## Example of JSON Path

For an API call returning content of type application/json, this sample could be part of the reply:

```
{
  "MIRecord": [ {
    "NameValue": [
      {
        "Name": "CONO",
        "Value": "780"
      },
      {
        "Name": "DIVI",
        "Value": " "
      },
      {
        "Name": "LNCD",
        "Value": "GB"
      },
      {
        "Name": "CUNO",
        "Value": "AA001      "
      },
      {
        "Name": "CUNM",
        "Value": "Swedish Customer Ltd-CHANGED-----]"
      },
      {
        "Name": "CUA1",
        "Value": "Mosstorpsv 32, Täby      "
      },
      {
        "Name": "CUA2",
        "Value": "      "
      },
      {
        "Name": "CUA3",
        "Value": "16441 KISTA      "
      },
      {
        "Name": "CUA4",
        "Value": "Sweden      "
      }
    ]
  }
]
```

To retrieve the first Value element, you can use this JSON path:

```
$.MIRecord[0].NameValue[0].Value
```

The resulting value in this case is "780".

If the path points to a list of values, the first occurrence is by default selected.

To indicate a specific element of an array, use an index. Counting starts from 0, so `MIRecord[0]` also returns the first element.

To retrieve the value for the customer name (CUNM), this JSON path with a filter can be used:

```
$.MIRecord[0].NameValue[?(@.Name=="CUNM")].Value
```

The resulting value in this case is "Swedish Customer Ltd-CHANGED-----"]".

## Example of an XPath

For an API call returning content of type `application/xml`, this sample could be part of the reply:

```
<?xml version="1.0"?>
<SalesOrder>
  <SalesOrderHeader>
    <CustomerParty>
      <PartyIDs>
        <ID
          accountingEntity="customer180"
          lid="lid://infor.app.nldv1180">RGT000043</ID>
        </PartyIDs>
        <Name>Purchasing Business Partner</Name>
        <Location type="Office">
          <Address>
            <AttentionOfName>Millers & Co.</AttentionOfName>
            <AddressLine sequence="1">De Dam 1</AddressLine>
            <AddressLine sequence="2"/>
            <CityName>Amsterdam</CityName>
            <CountrySubDivisionCode>NH</CountrySubDivisionCode>
            <CountryCode>NL</CountryCode>
            <PostalCode>1001 BA</PostalCode>
          </Address>
        </Location>
        <Location type="Billing">
          <ID
            accountingEntity="customer180"
            lid="lid://infor.app.nldv1180">D_RGSL1</ID>
          <Name>Regional Office</Name>
        </Location>
      </CustomerParty>
    </SalesOrderHeader>
  </SalesOrder>
```

To retrieve the city name where the customer's office is located, you can use an XPath expression with a filter on the `Location` attribute `@type`:

```
SalesOrder/SalesOrderHeader/CustomerParty/Location[@type="Office"]/Address/CityName
```

The resulting value in this case is "Amsterdam".

**Note:** The XPath expressions are run in the default namespace of the XML document. If there are explicit namespace prefixes used in the tag names, you can use standard XPath syntax to find the right value.

## Example of HEADER response

In output parameters mapping you can also specify a path for a mapping with response of type HEADER.

An API call could return a response Header similar to this sample:

```
"strict-transport-security": "max-age=31536000; includeSubDomains",
"x-response-time": "1435",
"x-frame-options": "SAMEORIGIN",
"p3p": "CP=Infor doesn't have any p3p policies.",
"content-type": "application/json",
"content-length": "3",
"connection": "Keep-Alive",
"date": "Tue, 14 Feb 2017 15:48:57 GMT",
"": ""
```

To retrieve a header value, use as path the name, for example:

```
content-length
```

The resulting value in this case is: "3".


## Running a workflow with an ION API step

A workflow that contains an ION API step can be triggered in the same way as other workflows. During the workflow execution, the API calls are made using the service account that is configured in each step.

An ION API call from Workflow is successful if these conditions are met:

- There is a response code and this code is greater than or equal to 200 and less than 400. Codes less than 200, and codes greater than or equal to 400, are interpreted as an error calling the API.
- The execution of the XPath or JSON Path gives no error. See the table for details.

This table shows the error handling when an incorrect path is specified in output mapping:

Situation	Generates runtime error	Behavior
The specified path is correct, but the value for this path is not found (null). This mapping is set as Optional (checked).	 No error	Workflow continues and the workflow parameter that is used in this mapping is not changed.

Situation	Generates runtime error	Behavior
The specified path is correct, but the value for this path is not found (null). This mapping is not set as Optional (unchecked).	✘ This situation generates an error at runtime.	Workflow continues or fails, depending on the settings.
The specified path is correct, but the value that is found for this path is an empty string. The workflow parameter that is used in the mapping is of compatible type String.	✔ No error	Workflow continues and the workflow parameter that is used in this mapping is set with the empty string "" value.
The specified path is correct. The value that is retrieved does not have the same data type as the workflow parameter in which it should be saved.	✘ This situation always generates an error at runtime, even if this parameter mapping is marked as optional.	Workflow continues or fails, depending on the settings.
The specified path is not correct: wrong syntax, for example, missing a closing bracket.	✘ This situation always generates an error at runtime, even if this parameter mapping is marked as optional.	Workflow continues or fails, depending on the settings.

If execution errors are encountered, these are handled according to the workflow configuration in the **How to handle errors** section on the **Settings** tab:

- If the **Continue workflow without changing parameters** option was selected, the workflow continues to the next step. The workflow parameter values, as they were before the ION API step, are used.  
Optionally, you can select the **Response status code** check box and specify a workflow parameter. If the API call fails, the error response code is saved in this workflow parameter and can be verified in the next step in the workflow.  
The error response codes:
  - 503 or 504 in case the API has retried the maximum number of times configured in the **Advanced Settings** section. The last error code from a series of unsuccessful retries is returned.
  - A workflow specific error code "-1" if there was an exception trying to connect to the ION API Gateway.
  - Any other code less than 200 or higher than 399 which the API could return.
- If the **The workflow fails** option was selected, the workflow gets status Failed. The remaining steps are not performed anymore. The reason for failure is visible in one of these places:
  - An error alert sent for business process errors, if this was configured in the **Error Reporting** section.

For details, see [Receiving alerts for business process errors](#) on page 438.

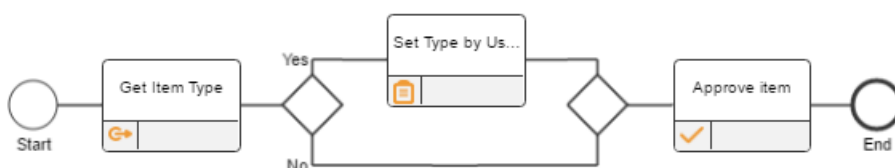
- The **Reason** column in the **Manage Workflow Instances** page in ION Desk. To open this page, select **Monitors & Workflows > Active Workflows**. Select the workflow name and click **Workflow Instances**.
- The `ProcessWorkflow.log` log file.  
In a cloud environment deployment, this log file must be retrieved by the Infor Cloud team.

## Avoiding workflow failure

A failed workflow cannot be continued or restarted. If some information is crucial for next steps in the workflow, but the workflow should not fail, you can use the following approach.

For example: an API call determines the type of a product and that type is required to determine who should approve this item. You can perform these actions:

- Assign an initial value, 'Unknown', to the Type parameter.
- Select the **Continue workflow without changing parameters** option in the runtime settings of the ION API step.
- After the ION API activity, add a decision flow to check the Type parameter. If it is still 'Unknown', send a task to a user to follow up. The user can, for example, look up the product type and specify it in the task.



## Using the optional output mapping

For some APIs, the response BODY may not contain all values for which a mapping is defined. In this case, to avoid that the ION API step results in failure, mark the output mapping for such values as optional.

In other cases, an API may return a different body in case of a successful call, and another body in case of an application error. In this case, create output mappings for all possible outcomes, but mark all as optional. After the ION API call, check the resulting parameters to decide if the call was successful or an application error occurred.

For example, assume this situation: the ION API call should determine the item type and, when an application error occurs, a different response BODY is received containing an error message. This table shows how you must define the output mappings to workflow parameters in this case:

Response	Path	Optional	Workflow Parameter
BODY	Path to item type element	Yes	Type
BODY	Path to error message	Yes	ErrorMessage

Before the ION API step, set the ErrorMessage to empty string (""). After the ION API step, add a decision step to check if an error message was received and let the user handle it.

You can use this feature in combination with the option to let the workflow fail in case of error. This way, you can avoid that the workflow fails in case of application errors.

## Procedures

### Task overview

In Workflow modeler and manage screens, you can perform these tasks:

- Create workflow definitions.  
You can create workflow definitions or import predefined workflow definitions.  
See these sections:
  - Creating a workflow definition
  - [Importing workflow definitions](#) on page 313
- Activate workflow definitions.  
After activation, a workflow definition is available for execution.  
See [Activating workflow definitions](#) on page 310.
- Change active workflow definitions.  
You can change some settings of an active workflow definition without deactivating the workflow definition.  
See "Changing active workflow definitions".
- Deactivate workflow definitions.  
After deactivation, a workflow definition is no longer available for execution.  
See [Deactivating workflow definitions](#) on page 312.
- Export workflow definitions to an XML file.  
See [Exporting workflow definitions](#) on page 313.
- Import workflow definitions from an XML file.  
See [Importing workflow definitions](#) on page 313.
- Analyze the status of workflow instances.



See these sections:

- [Viewing the status of workflows](#) on page 407
- [Viewing the status of a workflow](#) on page 407
- [Viewing the status of a workflow instance](#) on page 408
- View archived workflows.  
See [Viewing archived workflows](#) on page 410.
- Cancel running workflow instances.  
See [Canceling workflow instances](#) on page 317.

## Creating a workflow definition

- 1 Specify generic information for the workflow definition.  
See [Specifying generic information for a workflow definition](#) on page 273.
- 2 Specify the parameters to use in the workflow.  
See [Specifying workflow parameters](#) on page 274.
- 3 Show the usage of the workflow parameters.  
See [Showing the usage workflow parameters](#) on page 275.
- 4 Specify workflow drillbacks.  
See [Specifying workflow drillbacks](#) on page 275.
- 5 Specify workflow structures.  
See [Specifying workflow structures](#) on page 276.
- 6 Build the workflow model.  
See [Building the workflow model](#) on page 278.
- 7 Save the workflow definition.  
See [Saving the workflow definition](#) on page 280

## Specifying generic information for a workflow definition

- 1 In the **Workflows** modeling page, click **Add** to display the Workflow Modeler page. The Start and End nodes of the new workflow model are displayed on the canvas.
- 2 Specify this information:

### **Name**

Specify a unique name that identifies the workflow definition.

### **Description**

Specify the description that explains the functional purpose of this workflow.

---

## Specifying workflow parameters

- 1 In the left pane, click the [Show Workflow Properties](#) link to display the **Workflow Properties** pane. Alternatively, you can open the Workflow Properties pane by clicking the Start or End node of the flow. In the **Workflow Properties** pane, select the **Parameters** tab and click **Add** to start the **Workflow Parameter** window.

- 2 Specify this information:

**Name**

Specify the name of the parameter.

**Type**

Specify the data type, such as String or Boolean.

**Code Name**

This field is only displayed for parameters of the Code type. Specify the code list to use for the parameter. Codes are created in ION Desk.

See [Codes](#) on page 440.

**Group**

This field is only displayed for parameters of the User type. Specify the distribution group to use for the parameter. Distribution Groups are created in IFS. It is expected that only the person ids of IFS users that belong to the selected distribution group are assigned to the User parameter. By clicking the + (plus) sign a search on distribution groups is provided.

**Input**

If this check box is selected and a workflow instance is started. The initial parameter value is sent by the activation policy, or with the `ProcessWorkflow` BOD.

**Use Initial Value if Null**

If this check box is selected and an activation policy is used. The initial value is used if the required value is not available in the business object document that starts the workflow.

This setting is only used for activation policies. It is not relevant for workflows that are started using the `ProcessWorkflow` BOD.

This check box is only visible if **Input** is selected.

**Output**

If this check box is selected, the parameter is sent back in an `AcknowledgeWorkflow` BOD when a workflow instance is completed.

**Note:** If the **Input** and **Output** check boxes are cleared, the parameter is only used locally within the workflow definition.

**Initial value**

This field is enabled if you clear the **Input** check box or if you select **Use initial Value if Null**. Specify the initial value for the parameter. For the User and DistributionGroup types a search option is available.

- 3 Click **OK**. The new parameters are displayed in the **Workflow Parameters** pane.

**Note:** You can change the data type of a workflow parameter in these situations:

- The parameter is not yet used in a workflow step.

- The parameter is used only for display purposes in tasks, notifications, and task chain steps. Parameters are used for display purposes if they are included as a placeholder in the message or added to the step contents.

## Showing the usage workflow parameters

- 1 In the left pane, click the [Show Workflow Properties](#) link to display the **Workflow Properties** pane. Alternatively, you can open the Workflow Properties pane by clicking the Start or End node of the flow.
- 2 In the **Workflow Properties** pane, select the **Parameters** tab. A list of parameters is shown.
- 3 Select a parameter and click **Usage**. A list of workflow steps where this parameter is used is shown with their name and description. For *Decision* and *LoopBack* steps, the name of the *Used Condition* is displayed as step name.
- 4 Select one workflow step from the list and double-click to jump to the step properties pane. Here you can see how this parameter is used.

## Specifying workflow drillbacks

You can use a workflow drillback to create an Infor Ming.le drillback on a task or notification. The drillback is based on a view definition. The drillback contains a mapping from workflow parameters towards view parameters.

Like parameters, configured workflow drillbacks can be used on the task or notification **Content** tab to model the contents of the notification or task.

To add a drillback:

- 1 In the **Workflow Properties** pane, select the **Drill Backs** tab.
- 2 Click **Add** to start the workflow drillback window.
- 3 Specify this information:

### **Name**

Specify the name of this drillback link. This name is used to show the link in the task or notification.

### **Application**

Select an application name from the drop down list. These names are extracted from the drillback view definitions previously uploaded in the Infor Ming.le Admin Settings.

### **View**

Select a view from the list attached with the application selected before. The View parameters are added automatically in the lower part of the window.

### View parameters

There are three standard view parameters and one or more view-specific parameters. Each parameter can be filled with a predefined value, or mapped to a workflow parameter. To type in a predefined value, select the **Value** option. To map to a workflow parameter, select the **Parameter** option. You can select the placeholder <ACTIVITY\_ID> in the parameter drop down. This is replaced at runtime with the ID of the task or notification where this drillback is used.

This table shows the parameters:

Parameter Type	Description
Standard parameters	<p>These are the standard parameters:</p> <ul style="list-style-type: none"> <li>Logical ID - this must be filled with the Logical ID of the application to drill to. The value must start with <code>lid://</code></li> <li>Accounting Entity - this parameter is optional. It can be filled with a value of the accounting entity, or leave it blank.</li> <li>Location - this parameter is optional. It can be filled with a value of the accounting entity, or leave it blank.</li> </ul>
View-specific parameters	<p>Most often there is one parameter that identifies a document number, but a view could have any number of parameters. Consult the documentation of the application that delivered the Drill-back Views file about the meaning of each parameter.</p>

- 4 Click **OK**.

## Specifying workflow structures

Using workflow structures is optional, but after a workflow has structures, these are required input data.

To add a structure to a workflow definition:

- 1 In the **Workflow Properties** pane, select the **Structures** tab.
- 2 Click **Add** to open the **Structure** window.
- 3 Specify this information:

#### Name

Specify the name of this structure. This name is used to show the structure in the workflow steps where it is used. The name is also used as the root name of the structure when it is used to model input data for the workflow.

#### Label

Optionally, specify a label for the structure and add translations by clicking **Translate** next to the label input field. This label is displayed in the task and notification details.

See also [Adding translations](#) on page 328.

#### 4 Build the structure definition.

Hover over the structure node. Action buttons to add fields and child levels are displayed. The child level can contain fields and levels.

Complete these steps:

##### a Add a field.

Click **Add Field** on a level in the structure. A placeholder for a field is added as the last child element of the parent level.

Specify this information:

##### **Name**

The field name. This name is used when input data is specified for the structure and when the structure details are displayed in the manage screens and in the auditing reports.

##### **Label**

The label of the field to be displayed in the task and notification details. Optionally, add translations. Translated labels are displayed only in the task and notification details. In e-mail notifications, only the default labels are displayed.

##### **Type**

The data type for the values in this field. The input data must match this data type.

##### b Perform operations on the field.

Hover over the field definition. Action buttons to **Copy**, **Move**, or **Delete** the field are displayed. If you copy or move the field, placeholders are displayed for the locations where you can paste the field, or move the field to. When copying, you can paste the same field several times. To end the copy mode, click the white area in the **Structure** window. When moving, the placeholders disappear after the element is moved.

##### c Add a level.

Click **Add Level** on a parent level in the structure. A placeholder for a level is added as the last child element of the parent level.

Specify this information:

##### **Name**

The level name. This name is used when input data is specified for the structure and when the structure details are displayed in the manage screens and in the auditing reports.

##### **Label**

The label of the level to be displayed in the task and notification details. Optionally, add translations. Translated labels are displayed only in the task and notification details. In e-mail notifications, only the default labels are displayed.

##### d Perform operations on the level.

Hover over the level definition. These action buttons are displayed: **Add Field**, **Add Level**, **Copy**, **Move**, and **Delete**. The effect of these actions is similar as described above.

When the structure definition is complete, click **OK** to save it. From the workflow definition you can edit the structure and you can copy it, to the local workflow or to another workflow definition, using the **Copy** and **Paste** buttons.

## Building the workflow model

Use the Toolbox to add elements to the workflow model on the canvas. For each workflow element, specify its details in the corresponding properties pane. You can add these elements to the workflow model:

Category in Workflow Modeler Toolbox	Element
Activities	<ul style="list-style-type: none"> <li>• Task</li> <li>• Task Chain</li> <li>• Notification</li> <li>• Set Parameter</li> <li>• Decision Table</li> <li>• ION API</li> <li>• Wait</li> <li>• Start Workflow</li> </ul>
Flow	<ul style="list-style-type: none"> <li>• Decision</li> <li>• Loop Back</li> <li>• Parallel</li> <li>• Subprocess</li> </ul>

For each element to add:

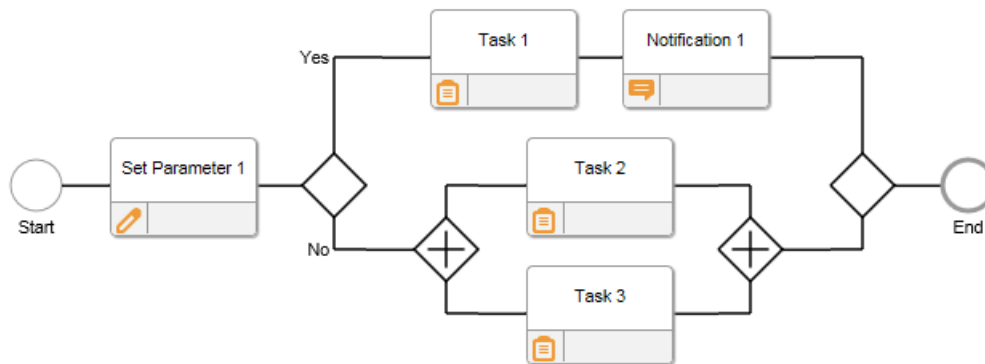
- 1 Select the element in the Toolbox and drag it towards the canvas. The possible drop locations are highlighted if you move the mouse pointer over the canvas. Drop the element at the desired location on the canvas.  
You can drag elements into a subprocess. If the subprocess is collapsed, click the Plus sign to expand it.
- 2 Specify the details for the element in the corresponding properties pane.

After you add a Loop Back element, you must add one or more elements between the start and end of the loop.

After you add a Decision or Parallel flow element, you must add one or more elements to each of its branches.

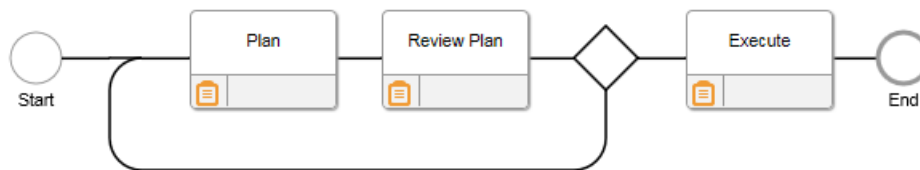
### Example 1: Workflow with Decision and Parallel Flow elements

For example, you add a Set Parameter element and a Decision element to an empty workflow. Then you add a task element and a notification element to the Yes branch. Then you add a Parallel element to the No branch of the Decision element. Finally you add task elements to the branches of the Parallel element. This diagram shows the resulting workflow model in the **Workflow Modeler**.



### Example 2: Workflow with Loopback element

If the review outcome indicates rework must be performed on the plan, then the workflow recycles. If the plan was OK then the workflow proceeds to the Execute activity. In this example the loop contains a sequence of tasks, but a loop back can include decisions, parallel flows or other loop backs. This diagram shows an example of a loop back in a workflow.



See these sections:

- [Specifying details for a task element](#) on page 280
- [Specifying details for a Set Parameter element](#) on page 289
- [Specifying details for a notification element](#) on page 288
- [Specifying details for a decision table element](#) on page 295
- [Specifying details for a task chain element](#) on page 284
- [Specifying details for an ION API element](#) on page 299
- [Specifying details for a wait element](#) on page 304
- [Specifying details for a start workflow element](#) on page 305
- [Specifying details for a decision element](#) on page 307
- [Specifying details for a parallel element](#) on page 307
- [Specifying details for a loop back element](#) on page 307
- [Specifying details for a subprocess element](#) on page 307
- Miscellaneous actions

**Note:** When saving the workflow definition, warning or error messages are displayed if:

- Elements are created that are not used in a flow.
- There are incomplete details for flow elements.

Solve the modeling errors before activating a workflow. Only valid workflows can be activated.

---

## Saving the workflow definition

Save the workflow definition and return to the Workflow Modeler overview page. The workflow definition is now complete. You can export the workflow definition for a local backup or for delivering it to customers.

See [Exporting workflow definitions](#) on page 313.

To deploy a workflow to the ION Service or to retract a workflow definition from being used, use the Activate/Deactivate functionality.

See these sections:

- [Activating workflow definitions](#) on page 310
- [Deactivating workflow definitions](#) on page 312

## Specifying details for a task element

- 1 Specify task properties.  
See [Specifying task properties](#) on page 280.
- 2 Specify task content.  
See [Specifying task content](#) on page 282.
- 3 Specify task actions.  
See [Specifying task actions](#) on page 282.
- 4 Specify task distribution.  
See [Specifying task distribution details](#) on page 283.
- 5 Optionally, specify task escalation and reminder settings.  
See [Configuring escalation and reminders](#) on page 328.
- 6 Optionally, specify task completion properties.  
See [Specifying task completion properties](#) on page 283.

## Specifying task properties

- 1 Select the Task element to display the **Task Properties** pane.
- 2 On the **Task** tab, specify this information:

### **Name**

Specify a name for the Task.

### **Description**

Specify a description for the Task.

### **Priority**

Specify the priority of the Task, such as High, Medium, or specify the message that is displayed as the task title in the Low.



## Message

Specify the message that is displayed to the users who receive this task. In the message, you can specify parameters between square brackets, which are substituted when the Task is distributed. You can specify a string up to 255 characters for the Task message or its translations.

When you press Ctrl+Spacebar, a list of available parameters is displayed. To select a parameter, use the arrow keys and press Enter, or use the mouse. To leave the list without selecting a parameter, press Esc.

For example, you can specify this message: Hello Purchasing Manager. Please approve Requisition [DocumentID] from Requester [RequesterName].

### Note:

- The parameters that are used in a message must be selected on the **Parameters** tab in the **Task Properties** pane.
- The parameter values of type string can contain up to 4000 characters. This can expand the length of the message. There is no technical limit on the number of parameters that are allowed in a message; although, depending on the resources available on the system used, an error may happen at runtime. We recommend that you use a maximum of 10 string parameters with placeholders in one message, if all these strings are potentially long.

You can use a special ## delimiter at the end of the Task message to define a category name. The string after ## is removed from the task message when this is displayed in Infor Ming.le. This string is listed in the Filter drop-down in the Task List widget configuration and can be used to filter tasks by category. If you use a parameter placeholder after ##, the value of the parameter is used as the category name. Comparison of category names is not case-sensitive.

You can specify a different message text for several languages. Click **Translation** next to the input text box. Specify a new message for each language that must be supported. To use categories, you must specify a category name in each message, for each language, using the ## delimiter.

## Due Date

A task due date is required to specify escalation and reminder rules at a time before or after the due date. Select one of the available options:

- **Time since task creation date**  
Specify a number and a time unit (days, hours, minutes). The due date is calculated automatically by adding this time offset to the task creation date and time.
- **Variable time since task creation date**  
This option is enabled only if the workflow contains at least one parameter of type integer. Select the parameter that contains the time offset and a time unit (days, hours, minutes). The due date is calculated automatically by adding this time offset to the task creation date and time.
- **Specified date and time**  
This option is enabled only if the workflow contains at least one parameter of type date and time. Select the parameter that contains the date and time value which must become the due date time of the alert.

**Note:** To see the due date in the email notifications sent for alerts, you must edit the template file `email_template.htm`. Remove the comments around the table row tag containing the `<_DUEDATE_TXT_>` placeholder.

---

See [Customizing email templates](#) on page 431.

## Specifying task content

On the **Content** tab, you can select which data to include in the task. You can also specify whether the task details should be displayed in an external form instead of in the Task List widget. To specify the task content:

- 1 Optionally, select the **Use custom form** check box and select one of the workflow drillbacks. In the Task List widget, this drillback link is executed when the user clicks to view the Task details.

Clear the **Use custom form** check box to use the standard task overlay to view the details.

**Note:** If the configured drillback is not correct, the user cannot complete the task from the ION **Task List** widget.

For more information, see [Workflow tasks with custom forms](#) on page 263.

- 2 The workflow parameters, drillbacks, and structures are displayed in the list on the left. To include some of these in the task, select the desired parameters, drillbacks, and structures, and click the right arrow button.
- 3 Set the label that is displayed in the user's Task screen in Infor Ming.le. You can specify translations for the label for several languages. To edit label translations, click the translation button next to the label field.

See [Adding translations](#) on page 328.

In case of structures, the label that is defined in the structure definition is reused. To change this label, you must edit the structure definition.

- 4 If a parameter must be displayed read-only for the user, select the **Read-Only** check box. Parameters having data type Hyperlink must be read-only, so that they are displayed as a clickable link. Drillbacks and structures are always read-only by default.

## Specifying task actions

The task actions are the buttons that users can click in the Infor Ming.le Task screen to complete a task.

- 1 Click the **Actions** tab.
- 2 Complete one of these steps:
  - Select **Use default action button 'Done'**. A **Done** button is displayed in the Infor Ming.le Task screen of the users to which the task is distributed.
  - Select **Use custom action buttons to set parameter** and select a parameter of type String, User, or DistributionGroup. Then add one or more buttons to be displayed in the Task screen. When one of these buttons is clicked, the task is closed and the value associated with that button is assigned to the selected parameter.
  - To add a button, click **Add Button** and specify this information:

### Button label

Specify the text that is displayed on the button. You can specify translations for button labels for several languages.

It is technically possible to specify a string with 255 characters length. The label text is not wrapped in the UI. Therefore, we recommend that you use short labels, for example, containing a maximum of 10 characters.

**Value**

Specify the value that is assigned to the parameter when a user clicks the button. For the User and DistributionGroup types a search option is available.

## Specifying task distribution details

A task or notification is distributed to the groups or persons displayed in the **Distribution** tab.

**1 Specify the **Distribution List**.**

Select one of these values:

**Simple**

A list is created by selecting elements one by one.

See [Adding distribution elements in simple configuration mode](#) on page 332.

**Advanced**

You can define a distribution matrix with conditional distribution rules.

See [Adding distribution elements in the advanced configuration mode](#) on page 334.

**2 Specify the **Distribution Type**. Select one of these values:****Create a single Task**

One specimen of the task is sent to the task queue of the selected distribution group. Only one user of the distribution group must handle the task.

**Create a parallel Task for each user in distribution**

All users in the distribution list receive a copy of the task in their Task List widget. Each user in the distribution list must handle the task.

## Specifying task completion properties

This step is optional and applies to steps of type task and task chain. Completion properties cannot be collected for parallel task.

You can specify task completion properties to collect this information:

- The user that completed a task
- The date and time the task was completed
- The final status of the task
- The Pulse ID that was generated for this task

To collect this information, you require workflow parameters of the applicable data type for each property.

**1 Click the **Completion Properties** tab.****2 Specify this information:**

**User**

If this check box is selected, you must select a workflow parameter from the list. The person ID of the user that completed a Task is stored in the selected workflow parameter. You can only select parameters of type String and User.

**Time Stamp**

If this check box is selected, you must select a workflow parameter from the list. The time information about when a Task was completed is stored in the selected workflow parameter. You can only select parameters of type Date And Time.

**Status**

If this check box is selected, you must select a workflow parameter of type String from the list. The final status of the task is stored in the selected workflow parameter. If this task is part of a task chain, then the status is from the last task in the chain. If the task was completed by a user, the value is DONE. If the task was automatically canceled by the system according to the configuration in the **Escalation and Reminders** tab, the value is CANCELLED.

**ID**

If this check box is selected, you must select a workflow parameter of type Integer from the list. The Pulse ID that was generated for this task is stored in the selected workflow parameter. If this task is part of a Task Chain, then the ID is from the last task in the chain.

**Note:** To return Task completion information when a workflow is completed, mark the parameters that are used to collect this information as output parameters.

## Specifying details for a task chain element

- 1 Specify task properties.
- 2 Specify task content.
- 3 Specify approval actions.
- 4 Specify the approval matrix.
- 5 Optionally, specify task escalation and reminders settings.
- 6 Optionally, specify task completion properties.

The task properties, content, escalation and reminders settings, and completion properties are the same as for a normal task.

See [Specifying details for a task element](#) on page 280.

For details on how to specify approval actions and the approval matrix, see the following sections.

## Specifying approval actions

Approval actions are comparable to task actions but a task chain activity always has two task actions, one to approve and one to reject.

To specify approval actions:

- 1 Select the task chain element. The **Task Chain Properties** pane is displayed.
- 2 Select the **Approval Actions** tab.
- 3 Select the parameter that must contain the result of the approval process (whether the users approved or rejected). This parameter must have data type String.  
**Note:** You cannot select this same parameter on the **Completion Properties** tab, because then the behavior of the task chain activity becomes unpredictable because one parameter can only be set with one value at a time.
- 4 Specify the label to be used for the approve button in the Infor Ming.le Task screen. For example, 'Approve'. If the task is distributed to people using different languages, specify translations for the label.
- 5 Specify the value to be used for the parameter when the Infor Ming.le user clicks the approve button.
- 6 In the same way, specify the label and value for the reject button.

## Specifying the approval matrix

The approval matrix defines which users should do the approval and in which sequence. A decision table is used, so you can define conditions that are based on workflow parameters. In this way the approval can, for example, depend on the amount that is involved. For a high amount, more people must approve than for a low amount.

There are two ways to specify the approval matrix to be used:

- To specify the approval matrix in the workflow modeler, select the **Simple approval matrix** option.
- To use a predefined approval matrix, select the **Use Business Rules** option.

### Using the "Simple approval matrix" option

To define the approval matrix in the workflow modeler:

- 1 Select the parameters to check. Parameters of type User and DistributionGroup are not supported.
- 2 Add rows to the matrix and define the conditions for each row.
- 3 Select the distribution for each row.
- 4 Change the sequence of the rows if required.

The first three steps are the same as for conditional distribution.

See [Adding distribution elements in the advanced configuration mode](#) on page 334.

In case of conditional distribution, a single task, notification, or alert is distributed based on all matching rows in the table. In the task chain, a task is distributed based on the first matching row in the table. If the user who handles the task approves, a new task is distributed to the next matching row in the table.

Therefore, the sequence of the rows in the approval matrix is important. You can change the sequence using the arrow buttons at the right side of the table.

**Note:** If the parameter values in a workflow do not match any row of the approval matrix, no approval task is created and the workflow will continue. The parameter selected on the **Approval Actions** tab will then have the same value as it had before the task chain activity.

### Example 1

This table shows an example of an approval matrix:

Parameters to check		Distribution
Type	Amount	
		Clerks
= 'Project'	> 100	ProjectManager
<> 'Project'	> 100	TeamManager
	> 1000	Director

The first Task is distributed to the people having the Clerks role. This Task is always created, independent of the Type and Amount. If the clerk who picks up the Task clicks **Reject**, the Approval Chain is completed. If the clerk clicks **Approve**, the conditions for the subsequent rows are checked.

For example:

- If Type = 'Project' and amount is 250, the approval sequence is: Clerks - ProjectManager.
- If Type = 'Training' and amount is 10,000, the approval sequence is: Clerks - TeamManager - Director.

### Example 2

In the previous example, the first row guarantees that at least one approval Task is created. The following table shows an example of an approval matrix where sometimes no Task is created. This can be used to automatically approve, for example if the amount is low.

Parameters to check		Distribution
Type	Amount	
= 'Project'	> 100	ProjectManager
<> 'Project'	> 100	TeamManager
	> 1000	Director

The parameter selected on the **Approval Actions** tab can have **Approved** as its initial value. In that case, if the amount is 100 or less, no approval Task is created, and the parameter keeps the **Approved** value.

### Example 3

To avoid that no approval Task is created, you can use a 'fallback' approver. In that case, complete these steps:

- Ensure that the parameter that is selected on the **Approval Actions** tab, such as Result, has a value other than the approved or rejected value. For example, **Initial**.
- Add the Result parameter as a parameter to check.

- At the end of the approval matrix table, add a row that checks whether the Result parameter still has its initial value.

This table shows an example:

Parameters to check			Distribution
Cost Center	Region	Result	
= 'c1'			CentralOffice
<> 'c1'	North		NorthOffice
<> 'c1'	South		SouthOffice
		= 'Initial'	Controller

A Task is created for the controller, for example, if the CostCenter is 'c2' and the Region is 'East'.

### Using the "Use Business Rules" option

Select this option to use an external approval matrix, defined in the Business Rules.

For details about creating and activating matrices see [Business Rules](#) on page 337.

To use an external approval matrix:

- Select the approval matrix name from the drop-down list box showing all active approval matrices. After a matrix is selected, its description is displayed next to the drop-down list box.
- After a matrix is selected, the list of input parameters for this matrix becomes available. These are displayed in a table with these columns:

#### Matrix parameter

The name of the matrix input parameter. Move the pointer over the name to see its description in a tooltip.

#### Data type

The data type of the matrix input parameter.

#### Workflow parameter

Select a workflow parameter to map to the matrix input parameter. The data type of the workflow parameter must match the data type of the matrix input parameter.

When mapping workflow parameters to matrix input parameters, the data type conversion is performed automatically. CODE, HYPERLINK, USER, DISTRIBUTION GROUP are converted to STRING.

### Migrating an approval matrix to Business Rules

You can migrate an existing configuration of an approval matrix to Business Rules.

- Go to **ION Desk > Monitors & Workflows > Workflows**.
- Open the workflow model that contains the approval matrix. If the workflow is active, you must deactivate it.

- 3 Select the `Task Chain` step and select the **Approval Matrix** tab.
- 4 In the **Task Chain Properties** panel, verify if the **Simple approval matrix** option is selected and click **Export**.
- 5 Confirm the matrix export to Business Rules when asked.  
In Business Rules, a new matrix of type `Approval Matrix` is created with the name `<workflow name>_approvalmatrix`. The migrated matrix contains the same parameters as the Simple approval matrix, the same names and the same data types as in workflow.
- 6 Review this matrix in Business Rules.
- 7 If required, change the matrix name and the parameter names before activating.
- 8 Submit the matrix for approval.
- 9 Approve and activate the matrix.

### Using the migrated approval matrix

To use the migrated approval matrix in the workflow model:

- 1 Go to **ION Desk > Monitors & Workflows > Workflows**.
- 2 Open the workflow model that contains the approval matrix. If the workflow is active, you must deactivate it.
- 3 Select the `Task Chain` step and select the **Approval Matrix** tab.
- 4 In the **Task Chain Properties** panel, verify if the **Simple approval matrix** option is selected and remove all rows.
- 5 Clear all parameters to be selected or set.
- 6 Select the **Use Business Rules** check box.
- 7 Select the migrated matrix from the drop-down list of the “Approval matrix”.  
This matrix must be approved in the Business Rules.
- 8 Specify the mappings for the matrix. Select the workflow parameters to be selected by the matrix conditions to map them to the matrix parameters.
- 9 Save and activate the workflow definition.

## Specifying details for a notification element

- 1 Select the notification element. The **Notification Properties** pane is displayed.
- 2 In the **Notification** tab, specify this information:

### **Name**

Specify a name for the notification. This name is displayed in the workflow model.

### **Description**

Specify a description for this workflow step.

### **Message**

Specify a message that the end user sees as notification. You can edit this message in the same way as the task message.

See [Specifying task properties](#) on page 280.



**Show notes from previous steps**

By default, this check box is not selected. The notification does not contain notes. If you select this check box, the notification does contain all notes from all previous tasks in the workflow.

**Expires after**

By default, a notification never expires. To avoid that too many notifications remain open if the users forget to close them, select the **Expires after** check box. Here you can specify an interval of time between one hour and 999 days. If the notification was not closed after this interval of time since it was created, the notification status is automatically set to Done. Expired notifications are automatically removed from the end user UI.

- 3 In the **Content** tab, select the parameters, drillbacks, or structures that must be displayed in the notification details.

The usage of notification parameters is similar to task parameters. For notifications, parameters are always read-only.

- 4 In the **Distribution** tab, specify the distribution list for the notification.

All users in the distribution list receive a notification simultaneously. To edit the distribution list, follow the same instructions as for a task distribution.

If the distribution list does not contain valid Infor Ming.le users when the notification is created, the notification status is automatically set to done.

See [Specifying task distribution details](#) on page 283.

## Specifying details for a Set Parameter element

- 1 Select the Set Parameter element to display the **Set Parameter Properties** pane.
- 2 Specify this information:

**Name**

Specify a name for the Set Parameter element.

**Parameters to be set**

Click the + icon to add one or more rows to the list of parameters to be set.

On each row, select a parameter name in the **Name** column. Each workflow parameter can be selected only once.

The **Data Type** column is automatically filled with the data type of the selected parameter.

The **Assignment Type** column is automatically filled with "Value" and the parameter is set with a default value specific to the data type.

Select a row to open the **Set Parameter** details panel. In this panel, you can change the **Assignment Type** and the values that are used in the assignment.

**Assign with**

Specify one of these assignment types:

- **Value:** The parameter is assigned with a fixed value.
- **Parameter:** The parameter is assigned with the value of another parameter.
- **Expression:** The parameter is assigned with an expression.

Only applicable for parameter types Integer, Decimal, Date And Time, String, and Hyperlink.

- **Person Name for Person ID:** The parameter is assigned with the value of the Common Name field corresponding to the given Person ID in IFS.

Only applicable for Parameter type String. The Person ID can be retrieved from a parameter of type String or User.

- **Value from Structure:** The parameter is assigned with a value extracted or calculated from a structure.

Only applicable for Parameter types String, Boolean, Integer, Decimal, Date, Date and Time.

### Value

This field is displayed if you specified the **Value** assignment type. Specify the fixed value that must be assigned to the parameter. This table shows an example:

Parameter	Assign with	Value
IsValid (boolean)	Value	True
IsValid (boolean)	Value	False

### Value of Parameter

This field is displayed if you specified the **Parameter** assignment type. Specify the parameter that must be assigned to the parameter. You can only select Parameters of the same type.

### Expression

This field is displayed if you specified the **Expression** assignment type. Specify the expression that must be assigned to the parameter.

For parameters of type integer and decimal, an expression consists of numeric parameters between brackets and operators. Valid operators are '+', '-', '\*', and '/'. Additionally, parentheses can be used.

To select from a list of available numeric parameters, press **Ctrl+Spacebar**. To select a parameter, use the arrow keys and press **Enter**, or use the mouse. To leave the list without selecting a parameter, press **Esc**.

This table shows an example:

Parameter	Assign with	Value
Profit (decimal)	Expression	[Salesprice] - [Purchaseprice]

You can assign a parameter of type integer or decimal with a value about the current workflow instance. These operators can be used:

- ID() to get the ID of the current workflow instance.
- MAJORVERSION() to get the major version of the workflow definition.
- MINORVERSION() to get the minor version of the workflow definition.

For parameters of type string and hyperlink, an expression consists of these elements:

- String constants
- String parameters
- Parameters of type hyperlink, code, or integer
- Operators

The parameters of type hyperlink, code, and integer are automatically converted to strings when the expression is evaluated.

You can use these operators:

- Ampersand (&) to concatenate two or more strings
- `SUBSTRING(string, start index, length)` to extract a substring
- `REPLACE(string, substring, replacement)` to replace all occurrences of substring
- `TRIM(string)` to trim beginning and ending white spaces. White spaces are space, tab, and enter
- `GETLAST(string, length)` to extract the last characters of a string
- `TOUPPERCASE(string)` to convert the whole string to uppercase
- `TOLOWERCASE(string)` to convert the whole string to lowercase
- `NAME()` to get the workflow definition name for the current workflow
- `SOURCETYPE(string)` to get the source type that started this workflow. Possible source types are:
  - Manual
  - Alert
  - Process.Workflow BOD
  - Workflow
  - Activation Policy

Note that when a workflow is started from a document flow the source type is Process.Wokflow BOD

- `SOURCE(string)` to get the identifier of the source that started this workflow instance. Depending on the source type, the identifier can be: the user identifier, the alert ID, the logical ID of the Process.Workflow BOD, the Activation Policy name or the workflow definition name..

You can nest the string operators. Use square brackets for parameter placeholders and double quotes for constant string values.

To select from a list of available parameters, press **Ctrl+Spacebar**. To select a parameter, use the arrow keys and press **Enter**, or use the mouse. To leave the list without selecting a parameter, press **Esc**.

The expression is validated when you leave the input field. If valid expressions are defined that fail at runtime, the workflow execution fails in this set parameter step. These are examples of failures:

- A parameter used in an expression has not been initialized and has value NULL.
- The result of a concatenation is a string with more than 4000 characters.

This table shows examples of expressions with strings:

Parameter	Assign with expression	Result
Prefix(string)	<code>SUBSTRING("abcde", 1, 3)</code>	"abc"
TestX(string)	<code>REPLACE("a_ba_ca_", "a_", "x")</code>	"xbxcx"
Link(hyperlink)	<code>"http://" &amp; [Prefix] &amp; ".com"</code>	"http://abc.com"
Address(string)	<code>[Street] &amp; SUBSTRING([AddressLine], 1, 3)</code>	Value of Street concatenated with the first three positions of the AddressLine

For parameters of type date and time you can specify an expression using these operators:

- NOW() to get the current date and time as GMT
- STARTTIME() to get the time when this workflow instance was started

### Person Name for Person ID

This field is displayed when you specified the **Person Name for Person ID** assignment type. Select a parameter that contains the person ID for which the user name should be retrieved. Any parameter of type string or User, other than the parameter to be assigned, is displayed and can be selected.

If the selected parameter contains a valid person ID, the result of this assignment is the value of the **Common Name** field from IFS, corresponding to this person ID. The person ID can be obtained from the completion properties of a previous user Task, or as an input parameter to the workflow. If the selected parameter does not contain a valid person ID, or there was an error retrieving the user name from IFS. The result of this assignment is the value of the parameter selected for the Person ID field.

Two users with the same name in a company can cause confusion. We recommend then to show the user name and the person ID in subsequent Tasks or Notifications.

This table shows an example:

Parameter	Assign with	Person Name for Person ID
UserName (string)	Person Name for Person ID	UserID (string, user)

### Function

This field is displayed if you specified the **Value from Structure** assignment type. Depending on the parameter data type, a pre-defined set of functions is available. You must select one function to apply. The function is applied to the element of the structure that is selected in the **Path** property. If the function does not return a valid result, the parameter is assigned with the **Default Value** specified.

This table shows the functions available for each data type:

Parameter type	Function	Result
Boolean	Get field value	Returns the field value that is specified in the path. Possible results are "True" or "False". If there are no occurrences of this field, the <b>Default Value</b> is returned.
Date or DateAnd-Time	Get field value	Returns the field value that is specified in the path. The result is a Date or a DateAndTime value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Maximum value	From all field occurrences that are specified in the path, returns the one with the highest value. This is the latest date value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Minimum value	From all field occurrences that are specified in the path, returns the one with the lowest value. This is the earliest date value. If there are no occurrences of this field, the <b>Default Value</b> is returned.

Parameter type	Function	Result
Decimal	Get field value	Returns the field value that is specified in the path. The result is a decimal value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Maximum value	From all field occurrences that are specified in the path, returns the one with the highest value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Minimum value	From all field occurrences that are specified in the path, returns the one with the lowest value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Average value	Calculates the average from all field occurrences that are specified in the path. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Sum of values	Calculates the sum of all field occurrences that are specified in the path. If there are no occurrences of this field, the <b>Default Value</b> is returned.
Integer	Get field value	Returns the field value that is specified in the path. The result is an integer value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Maximum value	From all field occurrences that are specified in the path, returns the one with the highest value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Minimum value	From all field occurrences that are specified in the path, returns the one with the lowest value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Count values	Counts all occurrences of the field or level that is specified in the path. If there are no occurrences of this field, the value zero is returned.
	Sum of values	Calculates the sum of all field occurrences that are specified in the path. If there are no occurrences of this field, the <b>Default Value</b> is returned.

Parameter type	Function	Result
String	Get field value	Returns the field value that is specified in the path. The result is an integer value. If there are no occurrences of this field, the <b>Default Value</b> is returned.
	Concatenate values	Returns the concatenated string of all occurrences of the field that is specified in the path. Delimited by the string that is specified in the <b>Separator</b> property. If there are no occurrences of this field, the empty string is returned. If the concatenated string values exceed a total length of 4000 characters, the execution of the workflow instance fails. An error message is displayed in the workflow instances manage screen.

### Path

This field is displayed if you specified the **Value from Structure** assignment type and is used in combination with the selected **Function**. You must specify a path in a structure on which the selected function is applied. To specify a path:

- a Click the zoom button (three dots) next to the **Path** field. A dialog box, where all workflow structures are listed, is displayed.
- b Expand the structure from which to make a selection. The structure fields and levels are displayed. Only the fields that have the same data type as the parameter in the assignment have a selection option next to them. Select an option to specify which field to use. At the bottom of the dialog, the path to the field is displayed. By default, the function applies to all occurrences of the field with this path from the workflow structure.
- c Optionally select one level from the path to the selected field. You can specify a filter for the level. This table shows the options:

Option	Effect
All	This is the default option. This means all occurrences of this level and its children are taken into consideration when the function is applied.
First	Only the first occurrence of this level and its children is taken into consideration when the function is applied.
Last	Only the last occurrence of this level and its children is taken into consideration when the function is applied.
Parameter	If you select this option, you must also select a parameter of type Integer. Set the specified parameter with a value that can be used as an index in the structure. Do this before the Set Parameter step is reached. Only the occurrence with the index that is specified in the integer parameter is taken into consideration when the function is applied. The first occurrence in a structure has index 1. If the index is out of bounds, the Default Value corresponding to the selected function is returned.

- d Click **OK** to confirm the path selection and return to the workflow model screen.

**Default Value**

This field is displayed if you specified the **Value from Structure** assignment type and is used in combination with these functions:

- Get field value
- Maximum value
- Minimum value
- Average value
- Sum of values

You must set this field to the value of the same data type as the parameter in the assignment.

**Separator**

This field is displayed if you specified the **Value from Structure** assignment type and used it in combination with the "Concatenate values" function for parameters of type String.

You can specify any string in this field. The specified string is used to delimit concatenated values. Empty string or spaces are also allowed as separator.

## Specifying details for a decision table element

Use a decision table step when a complex evaluation of several workflow parameters must be made to set values for one or more workflow parameters.

- 1 Select the decision table element. The **Decision Table Properties** pane is displayed.
- 2 Specify the decision table properties:

**Name**

Specify a name for the decision table. This name is displayed in the workflow model.

**Description**

Specify a description for this workflow step.

- 3 Specify the decision matrix.

You can specify the decision matrix to be used in these ways:

- Select the **Simple decision matrix** option in the workflow modeler.
- Select the **Use Business Rules** option, to use a predefined decision matrix that is exposed through the Business Rules.

## Using the "Simple decision matrix" option

- 1 Specify the decision matrix columns:

**Parameters to Check**

In this section, choose which workflow parameters must be used in the evaluation of the decision matrix conditions.

- a Click to open the **Parameters to Check** drop-down-list box. The list of workflow parameters is displayed.

- b Select one parameter. The parameter name is added as a column name to the decision matrix.
- c Repeat until all parameters to be checked are added.
- d To remove one column, click to open the **Parameters to Check** drop-down-list box and cancel the selection of the parameter name to be removed.

### Parameters To Set

In this section, choose which workflow parameters must be used in the assignment section of the decision matrix:

- a Click to open the **Parameters to Set** drop-down-list box. The list of workflow parameters is displayed.
- b Select one parameter. The parameter name is added as a column name to the decision matrix.
- c Repeat until all parameters to be set are added.

## 2 Add decision matrix rows:

- a Click the **+ ADD** button. An empty row is added.
- b Click the **Edit** icon button in each cell to edit its content.
- c To edit the Parameters to Check, a condition builder dialog with these choices is displayed:
  - Any Value
  - Parameter Comparison
  - Parameter Value Comparison

The condition types Parameter Comparison and Parameter Value Comparison are similar to the workflow conditions of this type. If the condition type Any Value is used, the comparison evaluates to True.

- d All conditions that are specified in the cells on the same row in a decision matrix are joined by logical AND implicitly. At evaluation time, the decision matrix row evaluates to True if all comparison conditions on the same row evaluate to True.
- e To edit the Parameters to Set, a dialog is displayed to specify the value for the parameter with the options:
  - Keep original value
  - Set value to...

To set the parameter to a specified value, you can only specify a value of the same data type as the parameter to be set.

- f Add more rows to the decision matrix to enter several conditions to be evaluated and corresponding values to be set. Combined conditions that are represented by each table rows are joined together with a logical OR at evaluation time.
- g To remove a selected row, click **Remove**.
- h Use the arrow buttons at the right side of the decision matrix to change the order of the rows. At evaluation time, the first decision matrix row that evaluates to True is sought.
- i The decision matrix evaluation result is the values for parameters to be set from the row that evaluated to True. If none of the rows evaluates to True, no changes are made to the values of the parameters to be set.
- j Save the workflow model to save the decision matrix configuration.



**Example:**

You create a decision matrix to calculate the shipping fee and whether insurance is required for sending a package, based on its weight and value. This table shows the parameters that are required for the workflow:

Parameter	Data Type
Weight	Decimal
Value	Decimal
Insurance	Boolean
ShippingFee	Decimal

This table shows a possible decision matrix for this workflow:

Parameters to check		Parameters to set	
Weight	Value	Insurance	ShippingFee
<= '800.0'	< '150.0'	False	3.5
> '800.0'	< '150.0'	False	12.0
	>= '150.0'	True	20.0

At evaluation time, these are possible results:

- For a package with Weight 700 and Value 120, no Insurance is required and ShippingFee is 3.5.
- For a package with Weight 500 and Value 170, Insurance is required and ShippingFee is 20.

## Using the "Use Business Rules" option

Select this option to use an external decision matrix, defined in the Business Rules.

For details about creating and activating matrices see [Business Rules](#) on page 337.

To use an external decision matrix:

- 1 Select the **Use Business Rules** option.
- 2 Select the decision matrix name from the drop-down list box showing all active decision matrices. After a matrix is selected, its description is displayed next to the drop-down list box.
- 3 After a matrix is selected, the list of input parameters for this matrix becomes available. These are displayed in a table with these columns:

### Matrix parameter

The name of the matrix input or output parameter. Move the pointer over the name to see its description in a tooltip.

### Data type

The data type of the matrix parameter.

**Input Workflow parameter**

If a matrix parameter is used in matrix conditions, this is an input parameter for the matrix. In this column, a drop-down is shown and you must select a workflow parameter to map to the matrix input parameter. The data type of the workflow parameter must match the data type of the matrix input parameter.

**Output workflow parameter**

If a matrix parameter is used in matrix columns for parameters to be set, this is an output parameter for the matrix. In this column, a drop down is shown and you can select a workflow parameter to map to the matrix output parameter. The data type of the workflow parameter must match the data type of the matrix output parameter. You can select the option “Ignore value” if you do not want to use this information in the workflow.

**Note:** When mapping workflow parameters to matrix parameters, these data type conversions are performed automatically:

- CODE, HYPERLINK, USER, DISTRIBUTION GROUP converted into STRING for matrix input parameters.
- STRING converted to CODE or HYPERLINK or USER or DISTRIBUTION GROUP for matrix output parameters.

## Migrating a decision matrix to Business Rules

You can migrate an existing configuration of a simple decision matrix to Business Rules.

- 1 Go to **ION Desk > Monitors & Workflows > Workflows**.
- 2 Open the workflow model that contains the decision matrix. If the workflow is active, you must deactivate it.
- 3 Select the `Decision Table` step.
- 4 In the **Decision Table Properties** panel, verify if the **Simple decision matrix** option is selected and click **Export**.
- 5 Confirm the matrix export to Business Rules when asked.

In Business Rules, a new matrix of type `Decision Matrix` is created with the name `<workflow name>_decisionmatrix`. The migrated matrix contains the same parameters as the simple decision matrix, the same names and the same data types as in workflow.

- 6 Review this matrix in Business Rules.
- 7 If required, change the matrix name and the parameter names before activating.
- 8 Submit the matrix for approval.
- 9 Approve and activate the matrix.

## Using the migrated decision matrix

To use the migrated decision matrix in the workflow model:

- 1 Go to **ION Desk > Monitors & Workflows > Workflows**.

- 2 Open the workflow model that contains the decision matrix. If the workflow is active, you must deactivate it.
- 3 Select the `Decision Table` step.
- 4 In the **Decision Table Properties** panel, remove all rows of the **Simple decision matrix**.
- 5 Clear all parameters to be selected or set.
- 6 Select the **Use Business Rules** check box.
- 7 Select the migrated matrix from the drop-down list of the “Decision matrix”.  
This matrix must be approved in the Business Rules.
- 8 Specify the mappings for the matrix:
  - a Select the parameters in the matrix conditions as "input workflow parameter".
  - b The parameters that are set when the matrix is executed must be set as "Output workflow parameter".
- 9 Save and activate the workflow definition.

## Specifying details for an ION API element

To model a workflow that makes an ION API call:

- 1 Start ION Desk and navigate to the workflow modeler.
- 2 Add a workflow step of type ION API.
- 3 ION API.

Click the **ION API** tab. Specify this information:

### **Name**

The workflow step name that is displayed in the workflow diagram. The same name is displayed in the Workflow Activity Locator. Define properties of the workflow step and prepare the call to the

### **Description:**

The step description that is used for modeling purposes.

### **Product**

The name of the application that exposes the ION API to call, as it was provisioned in Infor Ming.le ION API. The product name is selected through the operation selection dialog.

### **Operation**

The API operation to call.

Click **SELECT** to open the dialog that helps searching for an ION API operation using the ION API metadata. First select a product from the drop-down, then type in a search string and click the search icon. If the operation you are searching for is not in the list, click **Show More**. You can use the browser search function to refine the search within the resulting list of operations.

### **Service Account**

The service account obtained from IFS that grants permission to call the selected API operation. Click **IMPORT** to select the service account file and load it into the workflow step. After import, the description of the service account is displayed. You can use the same service account in several workflow steps. The credentials of this service account are used in these situations:

- When calling the API from ION DeskDefine properties of the workflow step and prepare the call to to perform a Test API Call.
- At runtime, when the workflow performs the ION API step.

**Note:** When exporting a workflow model that contains an ION API step, the service account that is used by this step is not exported. This is intentional, to prevent that service accounts are misused. After importing a workflow, you must reconfigure the service accounts in each ION API step.

#### 4 Specify the request parameters.

Click the **Request Parameters** tab. This tab is populated with the list of request parameters required to make the API call after an operation is selected on the **ION API** tab. These parameters are listed in a table with these columns:

##### Name

The operation parameter name as defined in the API metadata. An asterisk indicates whether it is required to pass a value for this parameter.

##### Description

The operation parameter description as defined in the API metadata.

##### Data Type

The operation parameter data type that was defined in the API metadata.

##### Parameter Type

Can be PATH or QUERY. Path parameters are used to build up the operation path and may not be null or empty. Query parameters are used to pass on additional values and may be optional or empty strings.

##### Value

To specify a constant value to use in all API calls for the API parameter on this row, select the radio button in this column.

##### Workflow Parameter

Select a workflow parameter to pass a value from the workflow to the API operation parameter when calling the API. This is required only when the API operation parameter is required. If you do not select a workflow parameter, then the null value is used in the mapping.

##### Note:

- You can map either a workflow parameter, either a constant value to the request parameter of an API operation. The radio buttons in the **Value** and **Workflow Parameter** columns help to enforce this choice for each row. The default selection is workflow parameter.
- The mapped parameters or values are used at runtime as input values to call the API. All workflow parameters are converted from any data type to string. This table shows the conversions that are performed by default:

Data type	Formatting
Integer and Decimal	Decimal separator is dot "." There is no number grouping character
Strings (including User, Group, Hyperlink, Code)	Space is encoded as %20 Values are not trimmed

Data type	Formatting
Boolean	Expressed as "true" or "false"
Date and DateAndTime	UTC format is used YYYY-MM-DDTHH:MM:SSZ (example: 2011-10-24T22:46:00Z)

## 5 Specify the request body.

Click the **Request Body** tab. This tab contains the properties of the request details of type body. If the selected operation does not have a request body, the controls on this tab are disabled.

If the selected operation has a request body, this information is available:

### Name

The name of the parameter of type body that was defined in the API metadata.

### Description

The name of the parameter of type body that was defined in the API metadata

### Content type

Here you can select the content type you would like to send as request. If the API metadata specifies that this API operation can accept several content types, these are listed in a drop-down. Possible values are application/json, application/xml, text/xml, and text. If the API can accept only one content type, this is pre-selected and cannot be changed.

### Body

This is an input text box that must be filled in with data that is consistent with the content type. If the request body is required for this operation, you may not leave this text box blank.

If the request body is optional you can choose to perform these actions:

- Fill in the **Body** text box with valid data. This information is used as request body for the API call.
- Leave the text box blank. An empty string value is used as request body for the API call.

If the model schema is described in the ION API metadata, you can generate an example body. To fill the request body with a sample, click **EXAMPLE**. This button is available for these request content types:

- application/xml
- application/json

You can use parameter placeholders in the data for the request body. To specify a placeholder, press **Ctrl+Space** or **Ctrl+Alt+Space**. A list of workflow parameter names, structure names, and the `$notes()` placeholder is displayed, from which you can select one entry at a time. Alternatively, use these placeholders:

- `$value(<parameter name>)`

Use this syntax to insert a placeholder for a workflow parameter. The `<parameter name>` must be an existing workflow parameter. At runtime, the placeholder is replaced with the value of this parameter, converted to string. The same data type conversions are applied as for request parameters.

- `$escape($value(<parameter name>))`

Use this syntax to escape special characters in a string or in the value of a workflow parameter, for example `$/,/”,<, >, *, &, <` and so on.

- `$structure(<structure name>)`

Use this syntax to insert a placeholder for a workflow structure. The `<structure name>` must be an existing workflow structure. At runtime, the placeholder is replaced with the value of this structure.

- `$notes()`

Use this syntax to insert a placeholder for the notes collected in the user tasks that have been executed up to the current step in the workflow. At runtime, the placeholder is replaced with all the notes listed in a predefined structure.

**Note:** The `$structure(<structure name>)` and `$notes()` placeholders should be used only in the request body of a POST or PUT method. These assumptions are made:

- The method accepts a request BODY of the application/json or application/xml content type.
- The BODY contains an element that matches the definition of the workflow structures or the notes.

For structure definition and notes definition sample JSON files, see KB 2062292 on the Infor Support Portal.

If you are using parameter placeholders for the request body, these are listed as input parameters on the **Test** tab. Therefore, you can provide test values and validate the API call works with the specified request body.

If you are using structure or notes placeholders for the request body, these are not listed on the **Test** tab. When testing, sample data for the structures and notes is automatically generated.

## 6 Specify the output parameters.

Click the **Output Parameters** tab. On this tab you can map API responses to workflow parameters so they can be used later in the workflow.

On this tab you can select the Response Content Type if the API can return several types of responses. Possible values are application/json, application/xml, text/xml, and text. If the API can return only one content type, this is pre-selected and cannot be changed.

Here you can also specify the output mapping. Use the **ADD** and **REMOVE** buttons to specify as many mappings as required. The table for the output mappings has these columns:

### Response

Select from CODE, HEADER, BODY. These are API response types that may contain various information:

- CODE is a number which usually indicated if the API call was successful or not
- HEADER is a property list that contains header information like “content-type”
- BODY contains the API response data, usually as a JSON or XML structure. You can extract values from this type of response to map into workflow parameters using a Path.

### Path

When the response is a structure, specify a Path to extract a specific value. If the response BODY is a JSON message, you must specify a JSON Path. If the response BODY is an XML message, you must specify an XPath. For a response of type HEADER, the Path must be a header name. See “Specifying a path for output mapping” for details.

**Optional**

Select this check box if one output mapping may be skipped if the specified Path does not yield a value from the API response. See “Avoiding workflow failure”.

**Workflow Parameter**

Select the workflow parameter that contains the value that is extracted from the API response. The value that is extracted from the API response is converted to the workflow data-type.

**Note:**

- The Path is not required in these situations:
  - response type CODE
  - response type BODY when Response Content Type is text.
- The UI does not validate if the Path you typed is correct. Use the **Test** tab to verify if your configuration is correct.
- If you change the response content type after you defined output mappings, all mappings are removed.

**7** Test the API call execution.

Click the **Test** tab.

On this tab you can test the API call execution and the mappings to the workflow parameters. To use the test tab, all the previous configurations must be completed on the **ION API**, **Request Parameters**, and **Output Parameters** tabs.

The service account that is imported on the **ION API** tab is also used to perform the test call.

**TEST button**

Fill in values for the Input Parameters displayed on this tab and click **TEST**. If the API call is successful, the results are visible in the **Output Parameters** section on this tab. If the call did not succeed, an error is reported in a message dialog.

**Input Parameters tab**

Once mappings are completed on the **Request Parameters** tab, the workflow parameters used as input for the API call are listed here. If parameter placeholders are used in the request body, these are also listed. Specify values for these parameters to test the API call.

**Output Parameters tab**

After mappings are completed on the **Output Parameters** tab, the workflow parameters used as output for the API call are listed here. If the API call test is successful and the output mappings contain correct paths, the values for the output parameters are displayed here.

**Note:** The ION Desk time-out is applied while waiting for the API test call results. If the API does not respond within this time-out, the test fails. You cannot change the time-out configuration for the Test API call from ION Desk.

**8** Specify runtime settings.

Click the **Settings** tab. On this tab you can specify preferences for the execution of the ION API call at runtime.

There are two sections:

**How to handle errors**

Here you can choose how to let the workflow behave when the API call failed. You can choose for:

**Continue workflow without changing parameters.**

Choose this option if the API call can be ignored and the workflow must continue.

Optionally, you can select the **Response status code** check box and specify a workflow parameter of type string, integer or decimal. If the API call is not successful, the selected workflow parameter contains the response status code returned by the API.

**The workflow fails**

Choose this option if the workflow should not continue if the API call fails.

**Advanced Settings**

in this section you can modify the settings for the API call time-outs. There are two values:

**Retry Time if ION API is Unavailable (seconds).**

This is the period of time in which ION tries to restore the connection to the ION API Gateway. The default period is 600 seconds (10 minutes) in which ION retries to connect.

**ION API Call Timeout (seconds).**

This is the interval of time to wait for a response from the ION API call. The default value is 600 seconds (10 minutes).

- 9 Save and activate the workflow.

## Specifying details for a wait element

To specify that a workflow must wait for a specified amount of time during its execution, use a Wait step:

- 1 Add a Wait step to the workflow diagram at the place where the workflow execution must stop and wait for a specified amount of time.
- 2 Select the Wait element from the workflow diagram. In the properties pane specify step properties:

**Name**

Specify a name for this workflow step.

**Description**

Specify a description for this workflow step.

**Waiting period**

Select one of the available options:

- Wait for specified time interval. Specify a value between 1 and 999 and select time unit minutes, hours, days.
- Wait for variable time interval. Select a workflow parameter of type integer and select time unit minutes, hours, days.
- Wait until specified date and time. Select a workflow parameter of type date and time.

At execution time when the workflow reaches this step it waits for the specified amount of time and then the workflow with resume execution automatically. If waiting is specified until a date and time from the past, the workflow continues execution immediately.

During the waiting time the workflow instance status is **Running**.



---

## Specifying details for a start workflow element

Use a Start Workflow step to start another workflow. You can start any workflow except the current one. We recommend that the workflow to start contains at least one user task, especially in these situations:

- When you use this step in combination with a Loopback step.
  - When you use this step in a configuration where you let several workflows start each other in a cycle.
- 1 Add a Start Workflow step to the workflow diagram at the point where another workflow must be triggered.
  - 2 Select the Start Workflow element from the workflow diagram.
  - 3 Specify this information in the properties pane:

### Name

Specify a name for this workflow step.

### Description

Specify a description for this workflow step.

### Workflow to Start

Select the name of the workflow to start. Other workflows in this system are displayed with status active or inactive. If you click **REFRESH WORKFLOW** after a workflow is selected, the workflow properties about the input/output parameters are refreshed. See the related tabs **Input Parameters** and **Output Parameters**.

### Start type

- **Synchronous: wait for the started workflow to finish:** Wait for the completion of the started workflow and use the values of its output parameters in the current workflow.
- **Asynchronous: do not wait for the started workflow to finish:** Continue with the current workflow to the next step immediately after the other workflow is started.

Note: If starting the workflow failed, the current workflow continues with the next step, even if you selected to start the other workflow synchronously. See the information about the **Completion Properties** tab for hints about how to handle in this situation.

- 4 Select the **Input Parameters** tab to specify the input parameters for the workflow to be started.  
The list of input parameters of the workflow is displayed with their name, data type and initial value as configured in the workflow to start.  
For each mapping, select one parameter of the current workflow. Select a parameter with the same data type to pass on values from the current workflow to the workflow to be started.  
If an input parameter has an initial value, you can omit specifying a mapping by leaving the default selection to **Ignore value**. In this case, the workflow is started using the initial value of this parameter.  
**Note:** You cannot map workflow structures for the workflow to be started. In this case, a validation error is displayed in the workflow modeler. If you model and activate a workflow without structures, and you change this workflow and add structures later, the workflow fails to start. This error code is displayed: `NOT_STARTED_MAPPING_ISSUE`
- 5 The selected start type.

- You selected the start type **Synchronous: wait for the started workflow to finish** and the workflow to be started has output parameters.  
Collect the output values by mapping these to parameters of the current workflow. Select the **Output Parameters**. The list of output parameters of the workflow to be started is displayed with their name and data type. For each mapping, select one workflow parameter of the current workflow that has the same data type. After the execution of the started workflow, this parameter contains the resulting value.  
Keep the option **Ignore value** selected, to ignore the output values of the workflow to start.
- You selected the start type **Asynchronous: do not wait for the started workflow to finish**. If this workflow has output parameters, the output values are ignored. In this case you cannot collect the output values into the parameters of the current workflow.

## 6 Select the **Completion Properties** tab.

The properties on this tab are optional. With these properties you can evaluate if the workflow to be started has started or completed successfully. These options are available:

- Start workflow result:** If this check box is selected, you must specify a workflow parameter of type string that contains the result code. You can use a decision step in the current workflow to handle possible error situations after the Start Workflow step is executed.
- Started workflow instance id:** If this check box is selected, you must specify a workflow parameter of type integer that contains the instance id of the started workflow. If starting a workflow was not successful, this parameter is not changed.

This table shows the available workflow result codes:

Code	Description
STARTED	Returned for workflows started asynchronously to inform that the sub workflow was started successfully.
NOT_STARTED_NOT_FOUND	The workflow to be started was not active when this step was executed.
NOT_STARTED_MAPPING_ISSUE	Mapping to the workflow input and output parameters was not correct. This can happen if the sub workflow was changed in the meanwhile.
NOT_STARTED_SYSTEM_ERROR	(Sub) Workflow could not be started due to system error.
COMPLETED	Returned for workflows started synchronously; it means the sub workflow completed successfully.
CANCELED	Returned for workflows started synchronously; it means the sub workflow is canceled.
FAILED	Returned for workflows started synchronously; it means the sub workflow has failed.

---

## Specifying details for a decision element

- 1 Select the decision element to display the **Decision Properties** pane.
- 2 Click **Add** to start the **Condition Builder** window and add a condition to the decision element. For details on the available condition types, see [Workflow conditions](#) on page 257.
- 3 In the **Used Condition** field, select the condition.  
**Note:** You can add a combined condition to a decision element.  
See [Defining combined conditions for decision or loop back elements](#) on page 308.

## Specifying details for a parallel element

- 1 Select the parallel element to display the **Parallel Properties** pane.
- 2 Specify this information:  
**Continue flow when**  
Select one of these options:
  - **One branch completed**  
The workflow execution will continue when one branch has been completed. Any branch can be completed first, depending on the sequence of steps modeled on each branch. In this case the other branches are canceled as soon as all activities in one of the branches are completed.
  - **All branches completed**  
The workflow execution waits until all the branches that have been started by this parallel element have been completed.

## Specifying details for a loop back element

- 1 Select the loop back element. The **Loop Back Properties** pane is displayed.
- 2 Specify the maximum number of loops. The value must always be filled, to avoid endless loops.
- 3 Click **Add** to start the **Condition Builder** window and add a condition to the loop back element. The workflow will loop back if this condition evaluated to true. For details on the available condition types, see [Workflow conditions](#) on page 257.
- 4 In the **Used Condition** field, select the condition.  
**Note:** You can add a combined condition to a loop back element.  
See [Defining combined conditions for decision or loop back elements](#) on page 308.

## Specifying details for a subprocess element

- 1 Select the subprocess element. The **Subprocess Properties** pane is displayed.
- 2 Specify the name of the subprocess.

**Note:** You can collapse and expand a subprocess element.

See [Expanding and collapsing subprocesses](#) on page 310.

## Defining combined conditions for decision or loop back elements

The decision element or loop back element, for which you define a combined condition, must already contain at least two conditions.

To define a combined condition:

- 1** Start the **Condition Builder**. For example:
  - a** In the decision or loop back Properties pane in the **Workflow Modeler**, click **Add**.
  - b** In the **Conditions** tab in the **Activation Policy** detail page, click **Add**.
- 2** Specify this information:

**Name**  
Specify the name of the condition.

**Type**  
Select **Combined** from the list. The **Condition Builder** window shows the conditions that are already linked to the decision element, loop back element, or activation policy.

**Logical operator**  
Select **AND** or **OR**.
- 3** Compose the combined condition:
  - a** Select at least two conditions to combine.
  - b** Click **OK**.

### Example

The **Decision Properties** pane in the **Workflow Modeler** shows the conditions A, B, and C. You want to define this combined condition: A AND (B OR C).

Complete these steps:

- 1** In the **Decision Properties** pane, click **Add** to display the **Condition Builder** window.
- 2** In the **Name** field, specify **D**.
- 3** In the **Condition Type** field, specify **Combined**. The **Condition Builder** window shows the conditions A, B, and C.
- 4** In the **Logical operator** field, select **OR**.
- 5** Select conditions B and C.
- 6** Click **OK**. The new condition, D (B OR C), is displayed in the **Decision Properties** pane.
- 7** In the **Decision Properties** pane, click **Add** to display the **Condition Builder** window.
- 8** In the **Name** field, specify **E**.
- 9** In the **Condition Type** field, specify **Combined**. The **Condition Builder** window shows the conditions A, B, C, and D (B OR C).

- 10 In the **Logical operator** field, select **AND**.
- 11 Select condition A and condition D.
- 12 Click **OK**. The new condition E (A AND D) is displayed in the **Decision Properties** pane. You can now use the combined condition E in the Decision element.

## Miscellaneous actions

### Moving or copying elements in the modeler

You can use drag and drop to move items in the modeler:

- To move an activity, click the activity and drag it to the line where it must be moved to.
- To move a decision, loop back, or parallel flow, click the diamond shape at the beginning or the end of the item. Then drag the shape to the line where it must be moved to.

You can drag elements into a subprocess. If the subprocess is collapsed, hover over the subprocess to expand it.

You can also use copy and paste, or cut and paste:

- To copy or cut an activity, right-click the activity and select **Copy** or **Cut**.
- To copy or cut a decision, loop back, or parallel flow, right-click the diamond shape at the beginning or the end of the item. Then select **Copy** or **Cut**.
- To copy or cut a subprocess, right-click the subprocess box. Then select **Copy** or **Cut**.
- To paste an item, right-click the line where the item must be added and select **Paste**.

You can also use copy/paste or cut/paste to copy or move items from one workflow definition to another.

### Removing elements

To remove an element from the workflow model:

- 1 Right-click the element.
- 2 Select **Delete**.

**Note:** To remove a parallel flow, loop back flow, or decision flow, right-click the split element or the join element. To remove a subprocess, right-click the subprocess box. When you remove a parallel flow, loop back flow, or decision, all activities in the branches are also removed. When you remove a subprocess, all elements in the subprocess are also removed.

- 3 Click **YES** to confirm deletion.

### Adding additional branches to parallel elements

- 1 Right-click the parallel element.
- 2 Select **Add Branch**.

The new branch is displayed in the flow. You can add any activities or flow elements to the branch.

## Removing empty branches from parallel elements

- 1 Right-click the parallel element.
- 2 Select **Delete Empty Branches**.

**Note:** A parallel flow always has at least two branches. So you can only delete empty branches if more than two branches exist.

## Expanding and collapsing subprocesses

To expand a subprocess, click the **+** icon at the bottom of the subprocess box. Alternatively, right-click the subprocess and select **Expand**.

To collapse a subprocess, click the **-** icon at the bottom of the subprocess box. Alternatively, right-click the subprocess and select **Collapse**.

## Using the messages pane

The messages pane contains errors or warnings for the current workflow. For example, an error is reported if a task does not have a distribution.

Errors must be solved before you activate the flow. For warnings, you can choose whether they must be solved. To navigate to the involved item in the modeling canvas, click on the message line. The item is selected in the canvas and its properties are displayed in the properties pane.

## Activating workflow definitions

To activate new or deactivated workflow definitions:

- 1 In the **Workflows** modeling page, select one or more workflows to activate and click the **Activate** button.
- 2 The status of the selected workflow definitions is now **Active**. The workflows are ready to be started.

**Note:** If the status of a workflow definition is **Unknown**, ION Desk cannot connect to the ION Service to determine whether the workflow definition is active. In that case you cannot activate, deactivate, or edit the workflow definition.

## Changing active workflow definitions

You can change properties of workflow steps in a workflow model without deactivating the workflow definition. An overview of allowed changes is listed further in this section.

In case of other changes on the workflow definition, you must first deactivate it. No new workflows are started when the workflow definition is inactive.

To change a workflow definition without deactivation:

- 1 Open the workflow definition detail page and make the required changes.  
Here you can change the Workflow description and the labels of the structure elements.
- 2 Save the changes and return to the workflow definitions overview page.
- 3 Activate the changed workflow definition to apply the changes to the active workflow definition in the ION Service.

This table shows the changes you can make in the workflow steps without deactivating the workflow definition:

Workflow steps	Allowed changes
<b>User task</b>	Description, priority, message and its translations, option to show notes from previous steps, parameter labels and translations and read-only check box, distribution, escalation and reminders
<b>Task Chain</b>	<ul style="list-style-type: none"> <li>• Message and its translations</li> <li>• Content parameter labels</li> <li>• Translations and read-only check box</li> <li>• Approval actions labels and translations</li> <li>• Approval matrix add/remove rows</li> <li>• Add/remove parameters to check</li> <li>• Conditions and distribution in table cells</li> <li>• Escalation and reminders</li> </ul>
<b>Notification</b>	Description, message and its translations, parameter labels and translations, distribution.
<b>Set Parameter</b>	Parameter to be set and assignment type.
<b>Subprocess</b>	Name
<b>Loopback</b>	Number of loops
<b>Decision Table</b>	Description, add/remove parameters to set, comparison conditions in matrix cells, values for parameters to set, option to use Business Rules.
<b>ION API</b>	Description, Operation, Service Account, Input Parameters, Input Body, Output Parameters, Runtime Settings.
<b>Wait</b>	Description, configuration of time to wait.
<b>Start Workflow</b>	All properties can be changed. You can deactivate and change the workflow to be started when the main workflow is active.

**Note:**

- Features or workflow steps not mentioned in the previous table are not editable when the workflow is active.
- The changes for steps of type **User Task** or **Notification** are available after reactivation, for Tasks from running workflow instances that are not yet generated. The other changes are only available when a new workflow instance is triggered.
- Changes that involve parameters can use only existing workflow parameters.

- You cannot undo changes that are already saved.
- The Start Workflow step returns an error code if the configuration is not actual anymore because the workflow to be started is changed.

For the list of available result codes, see [Specifying details for a start workflow element](#) on page 305.

## Deactivating workflow definitions

You can deactivate active workflow definitions whose running instances are completed or canceled.

In the **Workflows** modeling page, select one or more workflow definitions to be deactivated and click the **Deactivate** button. New instances of these workflows can no longer be started.

## Checking usage of workflow definitions

To see where a workflow definition is used:






- 1 Select **Monitors & Workflows > Workflows**. Existing workflow definitions are displayed as tiles.
- 2 Move the pointer over the workflow definitions in which you are interested.
- 3 Click the [Usage](#) link in the bottom right corner of the tile.

A list of activations policies, document flows, and monitors using the selected workflow is displayed. To open the details page of the selected object, double-click the row or select the row and click the details icon on the blue panel.

You can only see the models for which you have view permissions. If you do not have view permissions for the Connect Modeling pages, the document flows that use the listed workflow are not shown. If you do not have permissions to view any of the modeling pages for activation policies, monitors, and document flows, the [Usage](#) link is not shown on the tile.

Alternatively, open the workflow definition model and click **USAGE** in the button bar on the details page.

This table shows the type icons:

Type	Name	Description	Status
	Data Lake Flow Name	Data Lake Flow Description	Data Lake Flow Status
	Document Flow Name	Document Flow Description	Document Flow Status
	Workflow Name	Workflow Description	Workflow Status
	Activation Policy Name	Activation Policy Description	Activation Policy Status
	Monitor Name	Monitor Description	Monitor Status



---

## Exporting workflow definitions

Workflow in ION has an export/import mechanism to perform these tasks:

- Backup and restore workflow definitions.
- Deliver workflow definitions to customers.

For example, Infor can export workflow definitions to an XML file. Infor customers can import this file into their Workflow environment.

To export workflow definitions:

- 1 In the **Workflows** modeling screen, select one or more workflow definitions to export and click **Export**.
- 2 Save the export file. You can now import the file into another Workflow environment.

## Importing workflow definitions

- 1 In the **Workflows** modeling screen click the **Import** button.
- 2 Select the file that contains the workflow definitions. You can now activate the imported workflow definitions.

## Workflow authorizations

Workflow authorizations are required to give users access to advanced features.

Features such as "Start Workflow" from Infor Ming.le and managing workflow tasks from the Task Manager widget for Homepages. In the **Workflow Authorizations** page you can map security roles from Infor Federation Services (IFS) to authorization types for workflow definition names from ION Desk

To start a workflow, a user must have "Start" authorization for the workflow, and the workflow definition must be active. When a user starts a workflow, the latest active version is started. . When configuring Workflow authorizations, the workflow definitions do not have to be active.

If users lose workflow authorizations, they can no longer start the workflows concerned. They can still view old instances of the started workflows as long as they have workflow authorizations to start at least one other workflow.

To manage tasks from a workflow in Task Manager, the user must have "Manage" authorization for this workflow. Tasks from all workflow instances of this workflow definition can be searched for in the Task Manager widget, using predefined search criteria.

To configure workflow authorizations with ION Desk:

- 1 In IFS., create one or more security roles that are assigned workflow authorizations.
- 2 In ION Desk, select **Authorizations > Workflow Authorizations**. The list of workflow authorizations that are already defined is displayed.

- 3 Click **Add** to grant authorizations for a workflow.
- 4 Select the authorization type **Start** or **Manage**.
- 5 Select a workflow name from the list of workflow definitions present in ION Desk.  
**Note:** Only workflow definitions that do not contain structures can be started from Infor Ming.le. Therefore, only this type of definitions is displayed in the list when the authorization type is Start.
- 6 Select a security role to grant authorization to users with this role.
- 7 Click **SAVE** to add or update this authorization configuration.  
For Infor Ming.le users, the authorization setting takes effect when they log on or refresh the Infor Ming.le application.
- 8 If you change the workflow by adding structures, you must remove the start authorizations for this workflow.
- 9 If you rename a workflow, the workflow authorizations are automatically updated to reflect the new workflow name. If you delete a workflow, the authorizations that are defined for this workflow are also deleted.
- 10 To remove authorizations, select one or more rows and click **Delete**.
- 11 To export authorizations, select one or more rows and click **Export**.  
The configuration is exported to an XML file.
- 12 To import authorization configuration, click **Import** from the toolbar.  
The imported authorizations are merged with the existing configuration and duplicate records are skipped.

## Starting workflow instances

Workflows are started by events outside the workflow system. An external application can start a workflow instance by one of these options:

- Send a Process.Workflow message that contains the workflow definition name and start parameters.
- Send a Sync message event that is received by a workflow activation policy.
- Send a message that is received by a workflow activity, as modeled in a document flow.
- Start a workflow from an alert, as modeled in an event monitor.
- Start a workflow from another workflow, using the Start Workflow activity.
- Start a workflow through the /workflow/start API from the /process/application endpoint.

Workflows can also be started manually by a user in Infor Ming.le.

When a workflow completes, its results are communicated in a response message to the application that sent the starting event. Depending on the start method, these messages are sent with the workflow results:

- If the workflow was started by a Process.Workflow BOD, an Acknowledge.Workflow message is sent with the values of the workflow output parameters.
- If the workflow was started by a Sync BOD through an activation policy, a Process message for the same Noun is sent. This Process message is sent with the values of the workflow output parameters. If the workflow does not have output parameters, the Process message is not sent.

---

If the Process message could not be delivered, a Confirm BOD is generated containing this Process BOD, so it can be resubmitted later.

- If the workflow was started from a workflow activity, as modeled in a document flow, the output of the workflow is added to the document. The document flow continues. If there is no next activity in the flow, the workflow output is ignored.
- If the workflow was started from an alert, the workflow output is ignored.
- If the workflow was started manually, the workflow status and its output parameters are updated in the Infor Ming.le UI.
- If the workflow was started synchronously from a another workflow, the output values are returned according to the output mappings defined.
- If the workflow was started through an API call, the workflow output is ignored.

## Starting workflow instances through ProcessWorkflow BODs

In various applications, users can start workflow instances. See the application's documentation.

To start a new workflow instance, the external application must send a ProcessWorkflow message containing the name of the workflow definition that is to be used. The ProcessWorkflow message can also contain other input parameters or input structures. When a workflow instance is complete, any output parameters are sent back in an Acknowledge Workflow message.

This section contains details about the message patterns that can be used for the Workflow BOD. For more details about the Workflow BOD contents, see the *Infor ION Development Guide*.

### ProcessWorkflow message patterns

If the ProcessWorkflow BOD is received and the request can be completed, the Workflow engine sends an AcknowledgeWorkflow message with `actionCode="Accepted"` and `status/code="Initial"` to the application that sent the ProcessWorkflow message.

If the request cannot be completed, the Workflow engine sends a ConfirmBOD and an AcknowledgeWorkflow message with `actionCode="Rejected"` and `status/code="Failed"`. The error message is displayed in the status/description.

When the workflow is complete, a final AcknowledgeWorkflow message is sent with `actionCode="Modified"` and `status/code="Completed"`.

If an error is encountered during the completion of the workflow, an AcknowledgeWorkflow message is sent with `actionCode="Modified"` and `status/code="Failed"`. The reason for the failure is displayed in the status/description.

If a workflow instance is canceled, an AcknowledgeWorkflow message is sent with `actionCode="Modified"` and `status/code="Canceled"`.

If an Acknowledge.Workflow BOD could not be delivered, a ConfirmBOD is generated containing this Acknowledge BOD, so it can be resubmitted later.

This table shows the message patterns of the `ProcessWorkflow` and `AcknowledgeWorkflow` messages. For details about the elements of the `Workflow` noun, see the *Infor ION Development Guide*.

Desired action	Sent <code>ProcessWorkflow</code>	Received <code>AcknowledgeWorkflow</code>	Workflow action
Start a new workflow instance	actionCode="Add", WorkflowDefinitionCode, All parameters defined as "input" ; all structures	actionCode="Accepted", Status/Code=Initial, DocumentID/ID, WorkflowDefinitionCode	A new workflow instance was started.
		actionCode="Rejected", Status/Code="Failed", Status/Reason	A workflow definition with this name is not Active or workflow parameters or structures do not match.
		actionCode="Modified", Status/Code="Completed", DocumentID/ID, , WorkflowDefinitionCode All parameters defined as "output"	Workflow is completed successfully.
Cancel a running workflow instance	actionCode="Change", Status/Code="Cancelled", WorkflowDefinitionCode, DocumentID/ID	actionCode="Modified", Status/Code="Failed", Status/Reason, DocumentID/ID, WorkflowDefinitionCode	Workflow was partially run, but execution failed before the End step was reached.
		actionCode="Modified", Status/Code="Cancelled", Status/Reason, DocumentID/ID, WorkflowDefinitionCode	The workflow instance was canceled.
		actionCode="Accepted", Status/Code=Cancelled, DocumentID/ID, WorkflowDefinitionCode	The workflow instance was canceled by the <code>ProcessWorkflow</code> BOD or through ION Desk.

Desired action	Sent ProcessWorkflow	Received AcknowledgeWorkflow	Workflow action
		<code>actionCode="Rejected", Status/Code=Failed, Status/Reason</code>	The Cancel Workflow request was refused. The reason for refusal is placed in the Status/Reason.

## Canceling workflow instances

Workflows can be canceled by external events or by manual intervention of an administrator. These are the options to cancel a workflow:

- By an external application sending a Process Workflow BOD with action *Change* and status *Cancelled*.  
See the application documentation and [ProcessWorkflow message patterns](#) on page 315.
- By a Workflow Activation Policy of type Cancel, when a Sync event has been received that matches the cancelation rule.  
See [Activation policies](#) on page 318.
- By the Administrator using the manage pages in ION Desk. This option should be used only as a fall back mechanism when it is required to control the system's behavior.  
See [Canceling a workflow instance in ION Desk](#) on page 317.
- By a user in Infor Ming.le, if the workflow was started by the same user. For details, see the Infor Ming.le documentation.
- If the workflow was started from another workflow it can be canceled when:
  - It is modeled on a branch of a join one-in step and this branch is canceled.
  - The main workflow is canceled.
  - The main workflow fails.
- If the workflow was started through the `/workflow/start` API call, it can also be canceled by the application that started it. To cancel the workflow, use the `/workflow/cancel` API from the `/process/application` endpoint. An application is identified by its logical ID.
- By a workflow schedule with an action to cancel a running workflow when a new instance is started.  
See [Workflow schedules](#) on page 323.

## Canceling a workflow instance in ION Desk

To cancel a workflow instance in ION Desk:

- 1 In the **Active Workflows** page, select the workflow definition to which the workflow instance belongs and click **Workflow Instances**.
- 2 In the workflow instances page, select the instance to cancel. You can cancel instances that have the **Running** status.
- 3 Optionally, select the check box to cancel all notifications generated by this workflow instance.

- 4 Click **Cancel**. The instance is stopped. The status changes to **Canceled**. All open tasks that are generated by the workflow instance are also canceled.

If the option to also cancel notifications was selected, all open notifications that are generated by the workflow instance are canceled.

## Canceling a workflow instance in ION Desk

To cancel a workflow instance in ION Desk:

- 1 In the **Active Workflows** page, select the workflow definition to which the workflow instance belongs and click **Workflow Instances**.
- 2 In the workflow instances page, select the instance to cancel. You can cancel instances that have the **Running** status.
- 3 Optionally, select the check box to cancel all notifications generated by this workflow instance.
- 4 Click **Cancel**. The instance is stopped. The status changes to **Canceled**. All open tasks that are generated by the workflow instance are also canceled.

If the option to also cancel notifications was selected, all open notifications that are generated by the workflow instance are canceled.

## Canceling running workflow instances in ION Desk

To cancel all running workflow instances of a workflow in ION Desk:

- 1 In the **Active Workflows** page, select the workflow definition to which the workflow instances to cancel belong and click **Workflow Instances**.
- 2 In the workflow instances page, optionally adjust the filter and click **Search**.
- 3 To cancel all running workflow instances from the resulting search range, click **Cancel All**.  
Optionally, select the check box to cancel all notifications. All open tasks that are generated by the canceled workflow instances are also canceled.

If the option to also cancel notifications is selected, all open notifications that are generated by the selected workflow instances are also canceled.

**Note:** Notifications can also be canceled from the **Monitors & Workflows > Activities** page.

## Activation policies

A workflow activation policy contains a monitoring rule that evaluates incoming Sync BODs, and a mapping feature to map BOD attributes to workflow parameters. Workflow instances are started, or canceled, when the rule is triggered. Only BODs from the same tenant as the activation policy are accepted.

If the activation policy is of the Start Workflow type, it starts a new workflow instance for each document received. In a default configuration, if a new instance of the same document is received, for which a workflow instance is still running, no new workflow instance is started. Optionally, you can choose to

cancel the running workflow instance and let the activation policy start a new workflow instance based on the latest document data. When the workflow is completed, the activation policy sends a Process message for the same Noun as the incoming Sync message.

If the activation policy is of the Cancel Workflow type, it evaluates the document identification and cancels the workflow instance that is running for the same document. The evaluation is based on Noun Name, Document ID, Accounting Entity, and Location.

## Defining activation policies

- 1 Specify the generic information for the new activation policy.
- 2 Select the document the policy will evaluate.
- 3 Select document attributes.
- 4 Optionally, create conditions to use in the policy's rule.
- 5 Specify the business rule that the policy must use to evaluate incoming documents.
- 6 Specify the workflow parameters and map the workflow structures.
- 7 Click **Save**. The activation policy is now complete. You can now activate the policy.

See the following sections for details.

## Specifying generic information for an activation policy

- 1 In the **Activation Policies** modeling page, click **Add** to display the **Activation Policy** detail page.
- 2 Specify this information:

### **Name**

Specify a unique name that identifies the activation policy. The name must contain the characters a-z, A-Z, 0-9 and '\_' (underscore). The maximum length is 255 characters. Multi-byte characters are not supported in the name of an activation policy.

**Note:** Activation policies and monitors are similar entities for the runtime engine. Therefore the activation policy names must differ from the monitor names.

### **Description**

Specify a description for the activation policy.

### **Policy Type**

Select one of these types:

- **Start Workflow**

Select this type if the policy must start a workflow.

A workflow instance is started each time a document is received for which the activation policy rule evaluates to True. By default, if a workflow instance is already running when an update of the same document is received, the document update is ignored. To cancel the running instance and restart the workflow for a document update, select **Cancel running Workflow**.

- **Cancel Workflow**

Select this type if the policy must cancel a workflow.

---

When a document is received, a workflow instance started by another activation policy might be running. If so, and the rule of the activation policy of the Cancel Workflow type evaluates to True, this workflow instance is canceled.

### Workflow

Select the workflow definition that the activation policy must start or cancel.

## Selecting the document the policy evaluates

- 1 On the **Document** tab in the **Activation Policy** detail page, click **Select Document** to display the **Select Document** window.
- 2 Select the document to be monitored.  
To filter the list of documents:
  - Select a type, when applicable, from the Type list.
  - Select a level, when applicable, from the Level list.
  - Specify a text in the **Filter** field.If the document type "BOD" is selected, you can also filter by document level "Custom" and "Standard". Filtering by level is not applicable for the types "JSON" and "ANY", as the documents of these types are always custom.
- 3 Click **OK**. The selected document is displayed in the **Document** tab.
- 4 Click **Save**.

## Selecting document attributes

Select at least one attribute of the policy's document. You can use the selected attributes in the policy rule, or the policy conditions.

- 1 On the **Attributes** tab, click **Add** to display the **Select Attributes** window.
- 2 Expand the tree and select the desired attributes. The selected attributes are displayed in the tooltip of the "Selected" count at the top of the window.  
For details on the attribute types you can select, see [Attribute types](#) on page 201.
- 3 Click **OK**. The selected attributes are displayed on the **Attributes** tab.
- 4 Optionally, change the names of the attributes on the **Attributes** tab. For example, you can specify the name of the document, to which the attributes belong, as a prefix.
- 5 Click **Save**.

## Creating conditions for an activation policy rule

- 1 On the **Conditions** tab, click **Add** to display the **Condition Builder** window.
- 2 Specify this information:



**Name**

Specify the name of the condition.

**Type**

Specify the condition type. Select the desired type from the list.

- In Activation Policies, you can use these condition types:
  - Attribute-Value Comparison
  - Attribute Comparison
  - Attribute Comparison With Calculation
  - Attribute Existence
  - Date and time check
  - Combined ConditionFor details, see [Monitor rule - condition only](#) on page 203.
- The condition types displayed in the list depend on the attributes that you selected for the policy.

- 3 Specify the remaining fields. The dialog box is dynamic: the displayed fields depend on the selected condition type.

For example: For a condition of the Attribute-Value Comparison type, you must fill in these fields: **Attribute**, **Operator**, and **Value**.

- 4 Click **OK**. The condition is displayed on the **Conditions** tab.
- 5 Click **Save**.

## Specifying the activation policy rule

- 1 In the **Rule** tab, specify this information:

**Rule Type**

Specify one of these rule types:

- **Condition only**

In this type of rule, you can only select one policy condition. To use multiple conditions, using AND or OR, you must apply a combined condition.

See [Combined condition](#) on page 259.
- **Value Change**

Use this type for example to monitor value changes within the same document. You can also use it to monitor whether a specific value is not reached in a given interval of time.

See [Monitor rule - value change](#) on page 213.

- 2 Specify the required information for the selected rule type. The displayed fields depend on the selected rule type.
- 3 Click **Save**.

---

## Specifying the workflow parameters and mapping the workflow structures

- 1 On the **Parameter Mapping** tab, map the workflow parameters to the corresponding attributes in the document. You must map all workflow input and output parameters to attributes that have the same data-type. You can map the attributes of type STRING to the workflow input parameters of type HYPERLINK, CODE, USER, or DISTRIBUTION GROUP.

**Note:** Be careful when selecting attributes that are repeating, or are part of a repeating structure in the BOD. Only one value is used in the workflow. For example, if you select an attribute from an order line, the BOD might contain multiple order lines that match the activation policy rule. In that case, the value from only one of the lines is used in the workflow. To send information from repeating attributes as input to the workflow, use structures.

- 2 On the **Structure Mapping** tab, map the workflow structures to attributes from the document. Completing the mapping for structures is optional. Structures are considered optional input data for the workflow. You cannot have structures as workflow output.

The structure root is mapped by default to the noun name from the document. You must map each level in the structure to a document attribute that has children in the document structure. Possible children are XML attributes or other XML elements. Click **Map** to select a document attribute to be mapped to the level. After a level is mapped, you can map its child fields to the child elements of the document attribute selected. To map a field, click **Map** next to the field name. A dialog box, that presents the document elements that have the same data type as the field you are mapping, is displayed. Select one element and click **OK**. The relative path from the parent level to the selected element is added to the mapping.

Optionally, specify a filter for the selected document element. Click the **Filter** button next to the path you mapped. The xml-attributes of the selected element and the xml-attributes of its repeating parent elements are displayed. To edit the filter, specify a value for one of the attributes and click **OK**. The path is updated with the filter. At runtime, only the occurrences that comply with the filter are extracted from the document and mapped into the workflow structure. If there are several occurrences of the document element mapped to one single field, only the first occurrence is selected.

**Note:**

- a If you make changes to the workflow structure in the workflow definition, refresh the configuration in the activation policy to retrieve the changes by clicking **Refresh Workflow** on the summary pane on the left. A refresh removes the existing mappings.
- b You can skip several levels when you map a field to a document element. This is called a "mapping with elevation". If you mapped several sibling fields using a mapping with elevation, we recommend that you use a filter to ensure the fields are selected from the same path determined by the filter.
- c Fields in workflow structures are single occurrence. Therefore, if the document element mapped to the field has multiple occurrences and you want to retrieve all occurrences, we recommend that you define a workflow structure level that is mapped to the direct parent of the document element. In other words, avoid mapping with elevation in this case.
- d You cannot map a flat structure from a BOD, such as a custom BOD, into a more complex workflow structure, with several levels.

## Activating activation policies

To activate new or inactive activation policies:

In the **Activation Policies** modeling page, select one or more activation policies and click the **Activate** button.

The status of the activation policies is now **Active**. The policies start evaluating incoming documents and starting workflows.

## Deactivating activation policies

In the **Activation Policies** modeling page, select one or more activation policies and click the **Deactivate** button.

The status of the selected activation policies is now **Inactive**. The policies stop evaluating incoming documents and will no longer start workflows.

## Exporting activation policies

Workflow in ION has an export/import mechanism to perform these tasks:

- Backup and restore activation policies
- Deliver activation policies to customers

For example, Infor can export activation policies to an XML file. Infor customers can import this file into their Workflow environment.

To export activation policies:

- 1 In the **Activation Policies** modeling page, select one or more activation policies to export and click the **Export** button.
- 2 Save the export file.

You can now import the file into another Workflow environment.

## Importing activation policies

- 1 In the **Activation Policies** modeling page, click the **Import** button.
- 2 Select the file that contains the activation policies and import.

You can now activate the imported activation policies.

## Workflow schedules

A workflow schedule is an automatic way to start workflow instances at regular intervals of time.

---

A schedule is a sequence of “runs”: each time the schedule timer issues a trigger, the workflow schedule attempts to start a workflow instance. Depending on the schedule configuration, a new workflow instance may be started for each “run”. In some situations, a “run” may pass without having started a workflow:

- If the workflow schedule is configured to skip if a workflow is already running
- If the workflow schedule is paused

For details about the execution of the workflow schedules, see [Manage workflow schedules](#) on page 412.

## Defining workflow schedules

- 1 Specify generic information for a workflow schedule.
- 2 Specify the schedule timing information.
- 3 Specify the schedule action.

See the following sections for details.

### Specifying generic information for a workflow schedule

- 1 In the **Workflow Schedules** modeling page, click **Add** to open the **Workflow Schedule** detail page.
- 2 Specify this information:

#### **Name**

Specify a unique name that identifies the workflow schedule. The name must contain the characters a-z, A-Z, 0-9, and \_ (underscore). The maximum length is 255 characters. Multibyte characters are not supported in the name of a workflow schedule.

#### **Description**

Specify a description for the workflow schedule.

### Specifying the schedule timing information

On the **Schedule** tab, specify this information to determine how often the schedule should run:

#### **Schedule Time Zone**

Select a time zone when the schedule should run. The default value is UTC. A schedule time zone is useful to start workflows at a specified time of the day according to your preferred time zone.

#### **Schedule Time Settings - Start**

Specify a start date and time. By default, this is the date when the workflow schedule is created, but you can also select a date and time in the future. The start date and time specify when the schedule should begin the evaluation of its time triggers.

The schedule's first run might be later than the specified start date. For example, suppose a schedule expression is specified as follows: “Run every first Wednesday of the month at 10:00 AM” and the

---

start date is 1st October. The schedule does not run on this date, unless the first of October is a Wednesday. It runs at the first applicable occurrence after the start date.

### Schedule Time Settings – End

By default, a schedule has no end date and it runs as long as it is active.

Optionally, to specify how a schedule should end, select one of these options:

- After multiple runs  
Specify the number of times a schedule should run. In this case, the schedule stops after the number of times a “run” has occurred according to the schedule expression. A “run” is also counted if the schedule has status “Paused” and it does not attempt to trigger a workflow. When a schedule reached the maximum number of runs, it has status “Completed” and it cannot be restarted anymore.
- End by  
Specify a date and time. The schedule runs according to the defined expression until the system has reached the specified end date-time according to the specified schedule time zone. When a schedule reached the end date-time, it has status “Completed” and it cannot be restarted anymore.

### Scheduling Details

You can specify a recurrence pattern for the schedule. The properties that you select here are used to define a CRON expression. This expression is used by the runtime to determine at which times the schedule should run. Only those time triggers are considered which are between the start date, or the current date, and the end date time, if specified.

For more details about CRON expressions, see: <http://www.quartz-scheduler.org/documentation/>.

### Schedule Preview

When you model the schedule details, you can click the **Show Preview** button from the **Schedule Preview** section. A sample of how the schedule will run with the selected options is displayed.

The **Show Preview** action shows the first three runs that would occur after the specified start date. In some situations, less than three runs might be displayed:

- If the end date is soon
- If the option to end “after a number of runs” is less than three

Use the schedule preview as a help tool to get insight in how the schedule will run. At the time of the activation, the schedule expression is calculated again according to the values last saved and the current time.

## Specifying the schedule action

On the **Action** tab, specify this information about the workflow that should be started by this schedule:

### Workflow to Start

Select the name of the workflow that should be started by this schedule. In the drop-down, all workflows are displayed, except for workflows that contain structures. At the time the workflow schedule is activated, you must activate the workflow separately.

### Trigger Workflow Instance

You can select one of these options about how to trigger a workflow instance:

- Run in parallel  
If this option is selected, the schedule triggers a new workflow instance every time it runs. This is the default option.

- **Skip running workflow**  
If this option is selected, the schedule determines whether there is already a workflow instance started from a previous run that has not yet completed. If another workflow instance is still running, the schedule does not start a new workflow instance.
- **Cancel running workflow**  
If this option is selected, the schedule determines whether there is already a workflow instance started from a previous run that has not yet completed. If another workflow instance is still running, the schedule cancels the old workflow instance and starts a new one.

### Parameter Mapping

After a workflow is selected, the workflow parameters table is filled with the list of workflow input parameters. You must specify values for the workflow input parameters to be used by the workflow schedule when starting a new workflow instance:

- If the workflow input parameters already have initial values, you can use these. You can also specify other values.
- If the workflow input parameters do not have initial values, you must specify constant values.

The workflow output parameters are ignored if the workflow is started by a workflow schedule.

## Activating workflow schedules

In the **Workflow Schedules** modeling page, select one or more workflow schedules and click **Activate**.

The status of the workflow schedules is now **Active**. If the schedule start date is in the past, the schedule starts executing with the next applicable moment in time. If the start date is in the future, the schedule starts executing with the next applicable moment in time that is after the start time.

## Deactivating workflow schedules

In the **Workflow Schedules** modeling page, select one or more workflow schedules and click **Deactivate**. The status of the selected workflow schedules is now **Inactive**.

Upon deactivation, all schedule events about how this schedule has run are deleted from the runtime. This schedule is not visible anymore in the **Active Workflow Schedules** page.

## Exporting workflow schedules

- 1 In the **Workflow Schedules** modeling page, select one or more workflow schedules to export and click **Export**.

**Note:** You can also export one workflow schedule from its details page.

- 2 Save the export file.

You can now import the file into another Workflow environment.

## Importing workflow schedules

- 1 In the **Workflow Schedules** modeling page, click **Import**.
- 2 Select the file that contains the workflow schedules and import it.

You can now activate the imported workflow schedules.

## Chapter 11: Monitors and Workflows advanced configuration

Some features are similar in monitor and workflow models.

For example these features:

- Adding translations for elements of a task, alert or notification.
- Configuring escalation and reminders.
- Creating a distribution list in simple or advanced mode.

### Adding translations

Configured messages and labels, which are visible to the end user in Infor Ming.le, can be translated. To translate a message or label, you must add text for a specific language.

- 1 Click the **Translate** button:



A window is displayed.

- 2 In the window, add texts for different languages.

In Infor Ming.le, the user will see the message corresponding to the language that is specified in the browser settings. If there is no text available for a specific language, the default text of the message/label is shown to the end user.

### Configuring escalation and reminders

You can specify several types of escalation rules and reminders for tasks and alerts. You can also specify that an alert, task or task chain can be canceled if they are not completed on time.

Escalation rules can be specified when a task or alert is not assigned or not finished in time after creation date. In this case the managers of the users in the current distribution list are informed. If the task or alert is not assigned at the time of escalation, the managers can assign it to themselves. Alternatively they can assign it to another user from the distribution list.



You can define several escalation levels. If no action is taken in the time interval specified in the escalation rule, the task or alert is escalated to the next management level.

Escalation and reminders can be specified when a task or alert is not finished before, after or at the time of its due date. For these types of events you must define an alternative distribution to users, groups, contacts, contact groups or emails. The manager hierarchy is not used in this case. When the escalation event takes place, the configured users are added to the distribution list. Then the users can take actions on this alert or task and receive future escalations. An escalation email is sent to these users:

- Users from the distribution list who have the `Send E-mail` flag turned on.
- Users that are configured as the distribution of the escalation rule.

A reminder email is sent only to the users, groups, or contacts that are configured for the reminder distribution. These users are not added to the distribution list.

**1** Click the **Escalation and Reminders** tab, to add a row for each escalation or reminder to define.

**2** On each row, you can edit the columns:

- **Event**
- **Action**
- **Distribution**

**3** Select one of these event types:

- **Not assigned in time after creation date**  
Specify a time interval. The task or alert is escalated if it is not assigned within the specified interval of time since its creation time.
- **Not finished in time after creation date**  
Specify a time interval. The task or alert is escalated if it is not marked as `Done` within the specified interval of time since its creation time.
- **Not finished in time before due date**  
Specify a time interval. The event that is specified in the **Action** column takes place if the task or alert is not marked as `Done` at the specified time before its due date.
- **Not finished on due date**  
The event that is specified in the **Action** column takes place if the task or alert is not marked as `Done` at the time of its due date.
- **Not finished in time after due date**  
Specify a time interval. The event that is specified in the **Action** column takes place if the task or alert is not marked as `Done` at the specified time before its due date.

**4** Specify settings for events not assigned in time after creation date and not finished in time after creation date.

The events **Not assigned in time after creation date** and **Not finished in time after creation date** preserve their functionality from previous releases:

- You can use each one of these events only once.
- The event action is **Escalation and email**. Emails are not sent if the **Send E-mail** flag is turned off.

- You cannot specify an alternative distribution for these events. In the **Distribution** column, specify up to ten management levels to which the task or alert must be escalated. The escalation rule is applied for each management level if a user for that management level exists in Infor Federation Services (IFS).

5 Specify settings for events not finished in time before/on/after due date.

For the events **Not finished in time before/on/after due date**, you can specify the **Action** and **Distribution** as described later.

In the **Action** column, you can select one of these options:

- **Only email reminder**

At the time when the specified event happens, an email reminder is sent to the users that are specified in the **Distribution** column. Emails are not sent if the **Send E-mail** flag is turned off.

- **Escalation and email**

At the time when the specified event happens, the task or alert is escalated. The users that are specified in the **Distribution** column are added to the distribution list. If the **Send E-mail** flag is turned on in the Pulse engine configuration settings and in the task or alert distribution list, the users from the whole distribution list receive an escalation email.

- **Re-assign and email**

At the time the specified event happens, the task or alert is assigned to the user that is specified in the **Distribution** column. If the **Send E-mail** flag is turned on in the Pulse engine configuration settings, this user also receives an email. This email informs the user about the alert or task being assigned.

The user that is specified in the **Distribution** column might not be part of the current distribution list of the task or alert. If so, the user is automatically added before the assignment.

If the alert or task was already assigned to another user, it is unassigned from this user and then assigned to the specified user. The user from whom the alert or task is unassigned receives an email; this email is only sent if the option to send email was enabled in the distribution configuration for this user.

- **Cancel alert, Cancel task or Cancel task chain**

At the time when the specified event happens, the alert or task is canceled. In case of a workflow task or task chain, the workflow step is considered completed, and the workflow continues to the next step. You can use the task completion property Status to determine if the workflow step was completed by a user or by a cancellation event.

In the **Distribution** column, you can specify one of these distributions:

- **Current Distribution**

This is the distribution list if the task or alert at the time of the event. A reminder sent to “current distribution list” for a task or alert that is assigned is sent only to the assignee.

- **User**

Select a user from IFS.

- **Group**

Select a group from IFS

- **Contact**

Select a contact from IFS.

- **Contact Group**  
Select a contact group from IFS.
- **E-mail**  
Specify an email address.

Additional notes for events not finished in time before/on/after due date:

- To use these events, you must specify a due date. For the time offset, you can specify a monitor attribute or workflow parameter.  
If the values of the monitor attribute or workflow parameter are not valid, this row is ignored; the evaluation continues for the next time interval.  
An error alert is created if **Error Reporting to Business Process Administrator** is enabled in the ION Service Configuration.
- 
- If a distribution element is not valid, the reminder or escalation email is sent to the rest of the distribution list. An error alert is created about the invalid distribution element if To combine different distribution types, you can use several events of the same type with the same time offset and different distribution. **Error Reporting to Business Process Administrator** is enabled in the ION Service Configuration.

## Adding distribution elements

Tasks, Alerts, and Notifications are distributed to users from a distribution list that is evaluated at runtime.

In a simple distribution configuration, the distribution list is created by adding users, groups, contacts, contact groups, e-mail, or attribute values one by one. At runtime, the elements of the distribution list are evaluated and all users or contacts in this list receive an Alert, Task, Notification, or e-mail.

In the advanced distribution configuration, you can specify conditional distribution, based on monitor attributes or workflow parameters. At runtime, the elements of the distribution list are gathered for the conditions that evaluate to True. There is also an option for default distribution if none of the conditions evaluate to True.

To add elements to a distribution list or conditional distribution list, the same **Distribution Data** dialog is used. All options in this dialog are available in both simple and advanced configuration mode.

**Note:** The sending of e-mail based on the distribution depends on the configuration of the ION Service.

If the send e-mail property for the Pulse Engine is not set to True, no e-mails are sent. To view or change the configuration, select **Configuration > ION Service**. For setting the SMTP server and From Address properties, contact your system administrator.

## Adding distribution elements in simple configuration mode

By default, the distribution list configuration mode is set to **Simple**. You can use action buttons to add, edit, duplicate, and remove elements in this list. To add elements, use the **Distribution Data** dialog box.

- 1 In the **Type** field, select one of these options:

### **Group**

If this option is selected, the message is sent to an Infor Federation Services (IFS) distribution group or a specified users list. All users in the group receive the alert, task, or notification in Infor Ming.le, and optionally an email notification.

### **User**

If this option is selected, the alert is sent to a specific user. Users are retrieved from IFS and must be active system users. A user receives an alert, task, or notification in Infor Ming.le, and optionally also an email notification.

### **Contact**

If this option is selected, the alert is sent to a contact from IFS. A contact is an external user who cannot access Infor Ming.le. A contact does not receive the alert in Infor Ming.le, but only receives an email.

This distribution option is available for alerts and notifications. It is not available for Tasks because contacts could not complete the tasks from outside the system.

### **Contact Group**

If this option is selected, the alert is sent to a contact group from IFS. All contacts in this group receive an email.

This distribution option is available for alerts and notifications. It is not available for Tasks because contacts could not complete the tasks from outside the system.

### **Email**

If this option is selected, the alert is sent to an email address.

This distribution option is available for alerts and notifications. It is not available for tasks because tasks cannot be completed from an email.

- 2 If you did not select **Email**, select one of these options:

### **Select from List**

Select this option to display the list of available users or groups retrieved from IFS. Select the groups or users to which the alert must be sent.

### **Use attribute or parameter value**

Select this option to display the list of available monitor attributes or workflow parameters. Select the attribute or parameter that contains a valid person ID or a valid group ID to which the alert must be sent.

For a User type distribution, you can select a parameter of type String or User. The String parameter must contain a valid IFS Person ID. For a Group Type distribution, you can select a parameter of type String or DistributionGroup. The String parameter must contain a valid Group name from IFS, or it can contain a list of IFS Person IDs with this syntax, case sensitive:

```
USERS (personID1;personID2;...)
```

**Note:**

- Incorrect values, non-existing person-IDs or distribution groups, are ignored.
- Do not use group names in IFS that start with "USERS(" and end with ")"
- If you use the syntax `USERS(list of person IDs)`, ensure that the IFS Person ID strings do not contain a semi-colon ";"
- A parameter or attribute value can have up to 4000 characters. An IFS Person ID field can have up to 256 characters in IFS. The values that you specify for the distribution must fit these limits, otherwise they are truncated and the distribution items are not valid

- 3** If you selected **Email**, select one of these options:

**Enter Email address**

Select this option to specify a fixed email address.

**Use attribute or parameter value**

Select this option to display the list of available monitor attributes or workflow parameters. Select the value that contains a valid email address to which the alert must be sent.

Depending on the mail server that you use, you can include multiple email addresses in the attribute or parameter. For example: 'someone@infor.com; someone\_else@infor.com'. The ION Service uses the contents of the attribute or parameter as the 'To' in the email. The ION Service does not check whether the address or addresses are valid for the email server that is used. So if the data is invalid, the sending or delivery of the email fails.

- 4** If you selected **User** or **Group**, optionally select these options:

**Distribute to the manager(s) of ...**

If this check box is selected, the alert is sent directly to the manager of the selected user or group. The manager of the users in the distribution list is determined automatically in the ION Service, based on information from IFS. If users in a group have different managers, all managers receive the alert.

**Send Email to the selected users or group**

If this check box is selected, the users receive an email in these situations:

- A task or an alert is created and sent to them.
- The task/alert is assigned to the user by another user.
- The task/alert is unassigned by another user.
- The task/alert is escalated.

- 5** Click **OK**.

The selected distribution information is displayed in the **Distribution** tab.

- 6** Optionally, add more elements to the distribution list.

For details on users and groups, see the *Infor Federation Services Administration Guide (ifsag)*.

## Changing the e-mail setting

For a distribution element of type **Group** or **User**, you can change the setting that determines whether an e-mail is sent.

- 1 Select the distribution element and click **Edit**. Alternatively, double-click the distribution element.
- 2 Select or clear the **Send Email to the selected users or group** check box.

## Adding distribution elements in the advanced configuration mode

To specify conditional distribution for an alert, task, or notification, first set the distribution list configuration mode to **Advanced**. In the advanced configuration mode, you can specify various conditions to be verified based on monitor attributes or workflow parameters, and which elements should be added to the distribution list based on conditions.

A distribution matrix contains these configuration areas:

- The evaluation part with conditions for parameters to check. To add conditions, first some monitor attributes or workflow parameters must have been defined.
- The distribution list part, with a distribution element for each conditional row.

To create the conditional distribution configuration:

- 1 Add columns to the distribution matrix:
  - a Click to open the **Parameters to Check** drop-down-list box. A list of available parameters is displayed.  
**Note:** For monitors, the drop-down list box is called **Attributes to Check**.
  - b Select the parameter to be added. A new column, with the same name as the parameter, is added to the table. Repeat this step to add all parameters to be checked.  
**Note:** For monitor attributes, only the attributes that have single occurrence can be used in a conditional distribution configuration.
  - c To remove columns, click to open the **Parameters to Check** drop-down-list box and cancel the selection of the parameter names to be removed.
- 2 Add rows to the distribution matrix using the **+ ADD** button.  
Optionally, change the order of the rows by selecting one row and using the **Move Row Up** and **Move Row Down** buttons. The order of the rows does not influence the runtime behavior.
- 3 On each row, edit the comparison conditions for each cell. A condition builder dialog with these choices is displayed:
  - Choices that are displayed in the Workflow Modeler:
    - Any Value
    - Parameter Comparison
    - Parameter Value Comparison
  - Choices that are displayed in the Monitors:
    - Any Value
    - Attribute Comparison
    - Attribute Value Comparison
    - Condition

The **Parameter Comparison/Attribute Comparison** and **Parameter Value Comparison/Attribute Value Comparison** condition types are similar to the workflow or monitor conditions of the same type. If the **Any Value** condition type is used, the comparison always evaluates to True.

You can use the **Condition** type to select a monitor condition that was previously defined for this monitor. You can select a condition of type **Attribute Comparison** or **Attribute Value Comparison** that is already defined for the attribute to check.

- 4 All conditions specified in the cells on the same row are joined by logical AND implicitly. At evaluation time, the distribution table row evaluates to True if all comparison conditions on the same row evaluate to True.
- 5 To set distribution elements, edit the last cell in each row. The **Distribution Data** dialog is displayed and you can select any distribution element. You can select only one distribution element for each row. After a type is selected, this cannot be changed later.
- 6 At evaluation time, all rows that evaluate to True in the distribution matrix are used to add elements to the distribution list.
  - You can have several rows with different conditions, each with a distribution element. In this case only the elements for the conditions that are True are added to the distribution list.
  - You can have several rows with the same conditions and different distribution elements. In this case all distribution elements are added to the distribution list when the condition evaluates to True.
  - If a row with all conditions of type Any Value is included in the distribution table, the distribution element on this row is always added to the distribution list.
- 7 Optionally, specify default distribution. This is used when none of the conditions evaluates to True.

**Note:** When using advanced distribution for monitors, you cannot use a monitor rule with multi-occurrence.

## Example

For a monitor, you configure conditional distribution based on the CustomerRegion attribute. Possible regions are EMEA, APAC, and US. For a Sales Order from a specific customer, you want to send an Alert to the SalesManager of the corresponding region.

To avoid using individual user names, you create, in IFS, a distribution group for each region:

- SalesManagerEMEA
- SalesManagerAPAC
- SalesManagerUS

This table shows a sample configuration with conditional distribution:

Parameters to be checked		Distribution list	
CustomerRegion	Type	Identifier	Send e-mail
= 'EMEA'	Group	SalesManager EMEA	Yes

Parameters to be checked		Distribution list	
CustomerRegion	Type	Identifier	Send e-mail
= 'APAC'	Group	SalesManager APAC	Yes
= 'US'	Group	SalesManagerUS	Yes












## Chapter 12: Business Rules


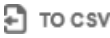



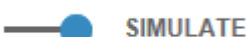
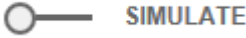
With the Business Rules application, you can manage the life cycle of approval and decision matrices.

Approval matrices contain rules, that are based on parameter evaluation, for distribution of approval tasks. These matrices can be used from Workflow to define approval chains. Decision matrices contain rules that are based on parameter evaluation to set values of output parameters. These matrices can be used from Workflow to define complex decision tables. One matrix can be reused in several workflows and can be updated without modifying the workflow definitions.

### Basic navigation

This table shows the icons used in the Business Rules application:

Icon	Description
	Business Rules
	Edit condition, parameter value or distribution
	Replace
	Note
	Submit for approval
	Undo Submit for approval
	Duplicate
	Revert
	Export

Icon	Description
	Import
	Export to CSV
	Import from CSV
	Approve
	Reject
	Simulate is switched on
	Simulate is switched off

## Permissions

The access to the Business Rules application is protected by ION Desk permissions and security roles. A user can be assigned to roles from Infor Ming.le™ User Management.

To see the new menu option **ION Desk > Monitors & Workflows > Business Rules** the users must have a permission of type "View" or "All" for one of the entries **Permissions > Monitors & Workflows > Business Rules > Editor or Approver** from Desk Permissions. To view and edit or approve matrices the users must have one of the `DECISIONSERVICE` Security Roles.

This table shows the `DECISIONSERVICE` roles:

Role	Functionality
<code>DECISIONSERVICE-Editor</code>	A user with this role can create and update a matrix and must submit it for approval before the matrix becomes active
<code>DECISIONSERVICE-Approver</code>	Users who can approve and activate or reject a matrix

A user who has both roles, `DECISIONSERVICE-Editor` and `DECISIONSERVICE-Approver`, has access to all available operations for a matrix.

## Permissions per role

This table shows the permission per role:

Action	Editor	Approver
Create new matrix, duplicate	Yes	
Edit matrix	Yes	
Revert Draft to last active version	Yes	
Import/export	Yes	
Import from/Export to CSV	Yes	
Submit for approval	Yes	
Approve and activate		Yes
Reject		Yes
Simulate execution	Yes	Yes
View all previous versions	Yes	Yes
View all matrices, filter	Yes	Yes

## Business Rules matrices

The `DECISIONSERVICE-Editor` and `DECISIONSERVICE-Approver` roles are required for working with approval and decision matrices.

An approval matrix is a collection of business rules based on input data to determine an approval chain sequence. A decision matrix is a collection of business rules forming a complex decision based on input data to determine values to send as output data. In the Business Rules application an Editor user can create a matrix and an Approver user can verify this definition and activate the matrix. After activation, a matrix can be used in workflow models.

## Lifecycle of a matrix

A matrix Editor can create and edit a new matrix.

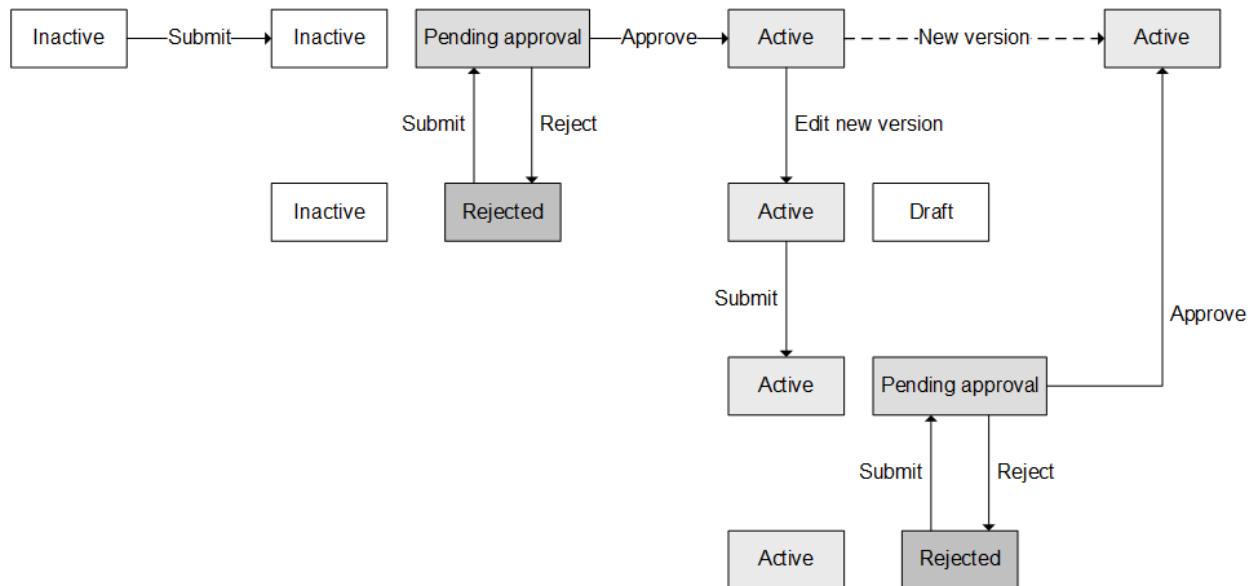
After the matrix is submitted for approval, the matrix cannot be edited anymore.

A user with an Approver role can approve or reject a matrix definition.

If a matrix is approved, a new version of this matrix becomes active. Each time a matrix is approved, its version number is increased.

If a matrix is rejected, the original version becomes editable again and after it is corrected, it must be submitted for approval.

This diagram shows the lifecycle of a matrix:



## Procedures

### Approval matrix definition

An approval matrix definition is composed of two areas:

- The input parameters  
Which together with the matrix name define the matrix interface. When this matrix is used in a workflow, values for all input parameters must be provided.
- The matrix rules  
Which is the set of rules to be evaluated on the input parameters to determine the distribution chain.

### Creating approval matrices

To explain how to create approval matrices, examples are available.

## Example 1

This table shows an example of an approval matrix:

Parameters to check		Distribution
Type	Amount	
		Clerks
= 'Project'	> 100	ProjectManager
<> 'Project'	> 100	TeamManager
	> 1000	Director

The first Task is distributed to the people having the Clerks role. This Task is always created, independent of the Type and Amount. If the clerk who picks up the Task clicks **Reject**, the Approval Chain is completed. If the clerk clicks **Approve**, the conditions for the subsequent rows are checked.

For example:

- If Type = 'Project' and amount is 250, the approval sequence is: Clerks - ProjectManager.
- If Type = 'Training' and amount is 10,000, the approval sequence is: Clerks - TeamManager - Director.

## Example 2

In the previous example, the first row guarantees that at least one approval Task is created. The following table shows an example of an approval matrix where sometimes no Task is created. This can be used to automatically approve, for example if the amount is low.

Parameters to check		Distribution
Type	Amount	
= 'Project'	> 100	ProjectManager
<> 'Project'	> 100	TeamManager
	> 1000	Director

The parameter selected on the **Approval Actions** tab can have **Approved** as its initial value. In that case, if the amount is 100 or less, no approval Task is created, and the parameter keeps the **Approved** value.

## Example 3

To avoid that no approval Task is created, you can use a 'fallback' approver. In that case, complete these steps:

- Ensure that the parameter that is selected on the **Approval Actions** tab, such as Result, has a value other than the approved or rejected value. For example, **Initial**.
- Add the Result parameter as a parameter to check.
- At the end of the approval matrix table, add a row that checks whether the Result parameter still has its initial value.

This table shows an example:

Parameters to check			Distribution
Cost Center	Region	Result	
= 'c1'			CentralOffice
<> 'c1'	North		NorthOffice
<> 'c1'	South		SouthOffice
		= 'Initial'	Controller

A Task is created for the controller, for example, if the CostCenter is 'c2' and the Region is 'East'.

## Decision matrix definition

A decision matrix definition is composed of two areas:

- The matrix parameters  
Together with the matrix name the matrix parameters define the matrix interface. When this matrix is used in a workflow, you must provide the values for all input parameters. The matrix output parameters can be mapped to workflow parameters, or can be ignored.
- The matrix rules  
A set of rules to be evaluated on the input parameters to determine values for the output parameters.

## Creating decision matrices

Create a decision table to calculate the shipping fee and whether insurance is required for sending a package, based on its weight and value. This table shows the parameters that are required for the workflow:

Parameter	Data Type
Weight	Decimal
Value	Decimal
Insurance	Boolean
ShippingFee	Decimal

This table shows a possible decision table for this workflow:

Parameters to check		Parameters to set	
Weight	Value	Insurance	ShippingFee
<= '800.0'	< '150.0'	False	3.5
> '800.0'	< '150.0'	False	12.0
	>= '150.0'	True	20.0

At evaluation time, these are possible results:

- For a package with Weight 700 and Value 120, no Insurance is required and ShippingFee is 3.5.
- For a package with Weight 500 and Value 170, Insurance is required and ShippingFee is 20.

## Defining a matrix

The DECISIONSERVICE-Editor role is required.

To define a matrix, first specify the matrix name and input parameters. Then add the matrix rules.

- 1 Click **Add** on the **Matrices** page to create a new matrix. Or open the details of an existing matrix to create a new version of this matrix.
- 2 When you click Add, you must choose the matrix type **Approval Matrix** or **Decision Matrix**. The corresponding detail page is displayed.
- 3 On the left side panel of a matrix, specify this information:

### Name

A matrix name can contain only letters, digits, and underscore “\_”. A name cannot be longer than 255 characters. After a matrix is approved and it becomes active in the runtime, its name cannot be changed anymore.

### Description

Explain the purpose of this matrix. A description can have maximum 4000 characters.

### Versions widget

To show information about the current and previous active versions of the matrix. For each version, this is shown:

- Status.
- Time when the matrix was last saved or approved.
- The name of the user who last saved or approved this version.

## Defining matrix parameters

Business data from application documents is sent to workflow as workflow input parameters. Workflow parameters are mapped to matrix parameters when a matrix is evaluated. The conditions that can be used for a parameter are specific to its data type.

Matrix parameters can be added or edited only when the matrix is created and not approved. After a matrix is activated, its interface, which is composed by the matrix name and its parameters, cannot be changed anymore.

Each parameter that is defined here must be used in one of these ways:

- As a column name or as a comparison parameter in a matrix condition.
- For an approval matrix, as a distribution option for “distribute to value from parameter”.
- For a decision matrix, as a parameter to be set.

You can only remove parameters that are not used in the matrix definition. To remove a parameter that is used, first remove the matrix column or rows where this parameter is used.

To define the matrix parameters

- 1 Click the **Parameters** tab in the right-hand panel.
- 2 Click **+Add** to add parameters to the list. Select one or more parameters and click **REMOVE** to remove them.
- 3 For each parameter, specify this information:

### Name

This name is visible in Workflow when this matrix is used and is used in workflow parameter mapping. The parameter name may have maximum 50 characters.

### Data Type

Select a parameter type from the list: STRING, INTEGER, DECIMAL, BOOLEAN, DATETIME, DATE, HYPERLINK, USER, GROUP or CODE.

Comparison condition operators specific to the data type can be used for each parameter. During workflow parameter mapping, only parameters with matching data-types can be mapped.

### Restriction

This property is applicable only for the data type USER or CODE. See the overview of the supported data types for details.

### Description

Explain the purpose of this parameter. You can use a maximum of 255 characters for the description. The description is visible during workflow parameter mapping.

This table shows an overview of the supported data types:

Data Type	Properties	Examples
STRING	A piece of text of maximum 4000 characters.	Used for names, codes, properties or descriptions
INTEGER	A whole numeric value of maximum 10 digits.	Used for whole numbers such as age, number of employees



Data Type	Properties	Examples
DECIMAL	A decimal numeric value, using a dot as decimal separator. A digit grouping symbol is not supported. Decimal values are not formatted according to the user locale settings.	Mostly used for amounts
BOOLEAN	Can be True or False.	Used for indicators such as Sellable item
DATETIME	A time and date value.	Used for dates, such as last updated date, planned delivery date
DATE	A date value. Time is considered as 12:00 AM.	Limited usage in application documents.
HYPERLINK	A string value that represents a hyperlink. The value must start with http://, https:// or ftp://	Mostly used for a website address such as <a href="https://www.infor.com/">https://www.infor.com/</a>
USER	A string value that represents a user identifier. Optionally, specify as Restriction a distribution group from Infor Ming.le User Management. When a Restriction is specified, only users who are part of the distribution group can be selected.	Used to identify a user. A valid identifier is an IFS Person ID of a user.
GROUP	A string value that represents the name of a distribution group from Infor Ming.le User Management.	Used to identify a group, such as TEST_USERS
CODE	A string value that represents a code from a Codelist. The Restriction property must be specified for this data type and must contain the name of a Codelist from ION Desk.	Used to ensure predefined values from an enumeration are reused consistently.

## Defining the matrix rules

To create the matrix definition:

- 1 Click the **Definition** tab in the right-hand panel.

- 2 Click **PARAMETERS TO CHECK** and select several parameters.

For each parameter, a column with the same name as the parameter name is added in the **Conditions** section. You can create comparison conditions for these parameters in each column. Columns are added in the same order as the order in which the parameters are selected. Optionally, reorder the columns using the right and left arrows. These arrows are displayed when you hover over the column name.

The **Actions** section contains the **Distribution** column in an approval matrix. You can edit the distribution for each row of conditions.

See [Editing the actions for an approval matrix](#) on page 348.

The **Actions** section contains the parameters to be set in a decision matrix. You can specify values for the output parameters for each row of conditions.

See [Editing the actions for a decision matrix](#) on page 351.

- 3 Click **+Add** to add rows to the matrix.
- 4 For each row, edit each cell and specify a condition.

Conditions that are specified on the same row are joined by logical AND for evaluation. If all conditions on a row evaluate to True, the distribution element that is specified for this row is added to the approval chain.
- 5 You can leave one cell blank. Consequently any value for this input parameter is accepted. In this case, this cell always evaluates to True.
- 6 Specify one comparison condition. Click inside a cell and specify an operator and a value. Use the conditions builder to get help with the syntax of various conditions:
  - a Click **Edit** at the left side of the cell to open the condition builder overlay.
  - b Select the condition type and specify the condition elements.
  - c To close the condition editor overlay, click the cell again or click outside the overlay.
- 7 Edit the matrix rows in these ways:
  - Select one row to apply these actions: Duplicate, Move Row Up, Move Row Down
  - Select one or more rows to apply the action: Delete
- 8 Save the matrix definition and re-submit the definition for approval.

## Overview of the condition types and their syntax

These are the condition types:

### **Any Value**

Applies to all data types.

The condition cell remains blank.

The condition evaluates to true for any parameter value.

### **Parameter Value Comparison**

Applies to all data types.

Compare the input parameter value with a constant value specified in the matrix.

Syntax: *Parameter Operator Value*

Select the operator from a drop-down list with operators specific for the data-type.

The value must be specified or selected and must be a valid value for the data type.  
This table shows operators for data types:

Data Type	Operator	Description
STRING	=	True if strings match, case-sensitive
	<>	True if strings do not match, case-sensitive
	contains	True if the parameter contains the specified string value
	notContaining	True if the parameter does not contain the specified string value
	startsWith	True if the parameter starts with the specified string value
	notStartingWith	True if the parameter does not start with the specified string value
	endsWith	True if the parameter ends with the specified string value
	notEndingWith	True if the parameter does not end with the specified string value
INTEGER, DECIMAL, DATE-TIME, DATE	=	Equals
	<	Less than
	<=	Less than or equal to
	>	Greater than
	>=	Greater than or equal to
BOOLEAN	<>	Does not equal
	=	Equals
	<>	Does not equal
HYPERLINK, USER, GROUP, CODE	=	True if strings match, case-sensitive
	<>	True if strings do not match, case-sensitive

### Parameter Comparison

Applies to all data types.

Compare the input parameter value with another input parameter

Syntax: *Parameter Operator Parameter*

The operator can be selected from a drop-down list with operators specific for the data-type. The same operators as in the Parameter Value Comparison apply.

Both parameters that are used in the comparison must have the same data type.

### Parameter Between Values

Applies to these data types: INTEGER, DECIMAL, DATETIME, DATE.

Compare the input parameter value with a start and end value. The comparison is including the specified values.

Syntax: *Parameter BETWEEN start value AND end value*

(Translates to 'start value' >= 'column parameter' <= 'end value')

### Parameter in Set

Applies to data type: STRING

Compare the input parameter value with a list of constant values, comma separated. The condition evaluates to True if at least one value matches the input parameter value. Comparison is case sensitive.

Syntax: *Parameter IN SET [Value1, Value2, Value3, ...]*

**Note:** Do not use spaces after the comma.

### Parameter not in Set

Applies to data type: STRING

Compare the input parameter value with a list of constant values, comma separated. The condition evaluates to True if none of the specified values matches the input parameter value. Comparison is case sensitive.

Syntax: *Parameter NOT IN SET [Value1, Value2, Value3, ...]*

**Note:** Do not use spaces after the comma.

### Parameter in Codes

Applies to data type: STRING

Compare the input parameter value with a list values selected from a Codelist. The condition evaluates to True if at least one code matches the input parameter value. Comparison is case sensitive.

Syntax: *Parameter IN CODES [CodelistName: Value1, Value2, Value3, ...]*

### Parameter not in Codes

Applies to data type: STRING

Compare the input parameter value with a list values selected from a Codelist. The condition evaluates to True if none of the selected codes matches the input parameter value. Comparison is case sensitive.

Syntax: *Parameter NOT IN CODES [CodelistName: Value1, Value2, Value3, ...]*

## Editing the actions for an approval matrix

The **Actions** area of an approval matrix is composed of the **Distribution** column.

Specify a distribution for each row in the matrix:

- 1 For each row, hover over the **Distribution** column and click the distribution icon to open the Distribution overlay.

- 2 In the overlay, select the distribution type and select one distribution element: user name, group name, or parameter name.
- 3 To close the distribution overlay, click the cell or click outside the overlay.

Two main distribution types exist:

- **USER:** For this distribution type, the approval task for that row of conditions is sent to one user. A user identification is based on the **IFS Person ID** property.
- **GROUP:** For this distribution type, the approval task is sent to a group of users. All users see the task, but only one user must complete it. The identification of a group is based on the name of a **Distribution Group** from IFS

This table shows the distribution options that you can use to specify a distribution to one user:

Type	Identification	Description
User	IFS Person ID	Select one user from the list of IFS users available. This user receives the task if the row of conditions evaluates to True.
User from parameter	Parameter name that contains the IFS Person ID	Select one of the matrix input parameters of type STRING or USER. This parameter must contain a valid IFS Person ID value when the matrix is called. If the parameter value is not a valid user identification, the approval task is generated, but it is not distributed to anybody. An administrator must re-distribute the task from ION Desk <b>Monitors &amp; Workflows &gt; Activities &gt; Task Details</b> .

These additional options are available in the distribution configuration:

- **Distribute to the manager of the selected user**  
If this check box is selected, the approval task is sent to the manager of the specified user. The manager is defined in IFS User Management from Infor Ming.le. If a manager is not defined, the approval task is generated, but not distributed to anybody. An administrator must re-distribute the task from ION Desk **Monitors & Workflows > Activities > Task Details**.
- **Send email to the selected user**  
If this check box is selected, the user receives an email about the approval task. If the **Distribute to the manager of the selected user** is selected, then only the manager receives the email. Emails are sent if the Pulse Engine email settings are configured and enabled.

This table shows the distribution options that you can use to specify a distribution to a group of users:

Type	Identification	Description
Group	Distribution Group Name	Select one group from the list available in IFS. The users in this group receive one common task if the row of conditions evaluate to True.

Type	Identification	Description
Group from parameter	Parameter name that contains the Distribution Group Name	<p>Select one of the matrix input parameters of type STRING or GROUP. When the matrix is executed, this parameter must contain a valid Distribution Group Name or a list of IFS Person IDs with this syntax:</p> <pre>USERS (personID1;personID2;...)</pre> <p>If the parameter value is not a valid group identification or a valid list of users, the approval task is generated but not distributed. An administrator must re-distribute the task from ION Desk <b>Monitors &amp; Workflows &gt; Activities &gt; Task Details</b>.</p>

These additional options are available in the distribution configuration:

- Distribute to the manager(s) of the users in the selected group**  
 If this check box is selected, the approval task is sent to the combined list of managers of each of the users in the group. The manager is defined in IFS User Management from Infor Ming.le. If no manager is found the approval task is generated, but not distributed to anybody. An administrator must re-distribute the task from ION Desk **Monitors & Workflows > Activities > Task Details**.
- Send email to the selected group**  
 If this check box is selected, an email is sent to all users in the group about the approval task. If the **Distribute to the manager(s) of the users in the selected group** is selected, then only the managers receive the email. Emails are sent if the Pulse Engine email settings are configured and enabled.

**Note:** At execution time, the rows are evaluated from top to bottom. An approval chain is constructed for all the rows for which the conditions evaluate to true, keeping the order from top to bottom.

## Search and replace

To replace a user in an approval matrix, you can use the Search/Replace feature. This feature applies to the latest version of a matrix and can be used on matrices that are active or inactive. This feature cannot be used while a matrix has the Pending Approval status.

To replace a user:

- 1 Open the details view of the matrix to edit and click the **Definition** tab.
- 2 At the upper-right corner of the matrix, click **Replace**.
- 3 Specify this information:

### Find User

The drop-down list in this field contains the user identifiers that are used in the **Distribution** column of the current matrix.

Select one value. The matrix view shows all the rows where distribution to this user is specified.

**Replace With**

Select the user identifier for the replacement.

- 4 Click **Replace** to apply changes, or click **Cancel** to abort this operation.
- 5 If you made changes, save and submit the matrix for approval.  
After the updated version of the matrix is approved, the replaced distribution user is used the next time the matrix is started from a workflow.

## Editing the actions for a decision matrix

The **Actions** area for a decision matrix is formed by one or more columns for parameters to be set. These parameters contain the output values after the matrix evaluation.

To add columns to the **Actions** area:

- 1 Click **PARAMETERS TO SET** and select one or more parameters.  
For each parameter, a column with the same name as the parameter name is added in the **Actions** section.
- 2 Specify values for these parameters in each column.  
The columns are added in the same order as the order in which the parameters are selected.
- 3 Optionally, reorder the columns using the right and left arrows. The arrows are displayed when you move the pointer over the column name.  
After the matrix is approved, you cannot add or remove columns from the **Actions** area.
- 4 For each row in the matrix, specify values for the parameters to be set:
  - By default, a blank cell is added for each column. This means that the parameter is not changed when this matrix row is executed. The parameter holds the same value as it had in the workflow, before the decision table was evaluated.
  - Click the edit icon in the cell to edit parameter values. Select one of these options:
    - **Keep Original Value**  
The parameter value is not changed. This is the default setting.
    - **Set Value**  
Specify the resulting value for this parameter. Only values that have the same data-type as the parameter to be set are allowed.

At execution time, the rows are evaluated from top to bottom. The Parameters to be set contain the values from the first rows for which the conditions row evaluates to true.

## Adding notes to matrix rows

To add a textual explanation about the conditions or the distribution on a specific row, you can add a note.

- 1 In the **Note** column, hover over the empty cell to see the **Note** icon.
- 2 Click the **Note** icon to open the Note editor overlay.
- 3 Specify a note text.
- 4 To close the overlay, click the **Note** icon again or click outside the overlay.  
The **Note** icon remains visible for rows for which a note is added. You can click the **Note** icon again to see the note text and to edit it later.

## Submitting a matrix for approval

After a matrix model is ready, you must submit the model for approval before it becomes active.

These actions are available:

- From the matrix detail view, click **Submit for approval**.  
Submit a matrix after all changes are saved. If validation errors are reported upon submission, you must fix the matrix and try again.
- If submitting was successful, the status has become **Pending Approval**.
- Optionally, click **Undo Submit for approval** to retract the approval request for the matrix.  
Use **Undo Submit for approval** before the approver approves or rejects this matrix. After **Undo Submit for approval**, the matrix status becomes **Draft** again and you can make more changes. After the changes, submit the matrix again.

## Duplicating a matrix definition

You can reuse the definition of an existing matrix with the Duplicate functionality.

- 1 Click **Duplicate** on the tile toolbar or on the matrix details page. A new matrix is created with the same input parameters and definition as the original matrix, starting with version 1.
- 2 Specify a new name for this matrix to make edits without restrictions.

## Reverting to the last active version

After a matrix is approved, you cannot change the approved version anymore. A new version with status **Draft** is created and is available to be edited and submitted for approval again.

- 1 To discard changes to a draft, click **Revert**.
- 2 Click **Save**. The matrix draft definition has returned to the last activated version.



---

## Importing and exporting matrix definitions

- 1 Open the matrix details page.
- 2 To save the definition of a matrix to a local file, click **Export**.  
To export several matrices to one file, select several matrix tiles on the **Matrices** overview page.
- 3 Open the **Matrices** page.
- 4 To import an exported matrix definition, click **Import**. An import creates a new matrix.

## Importing and exporting matrix definitions from and to CSV

You can export the definition of a matrix to a comma-separated values (CSV) file. You can edit the exported definition and then import it again. To use this feature, you must first add all matrix columns in the correct order. When exporting to CSV, the matrix definition is preserved. You can add rows to extend the matrix definition. When importing from CSV, the matrix that you import must match the matrix definition it is imported into.

### Exporting matrix definitions to CSV

- 1 Open the matrix details and click the **Definition** tab.
- 2 Click **Export to CSV** to export the current matrix definition to a CSV file.
- 3 Save the exported file. You can now edit this file and import the matrix again.

**Note:**

- An export is only performed if there are no errors in the matrix definition.  
This information is exported:
  - The **Notes** column.
  - The **Conditions** area with each column with its name and data type.
  - The **Actions** area with the **Distribution** column for approval matrices. Or each column with its name and data type for parameters to be set for decision matrices.
  - For each row in the matrix, the notes text, the conditions and parameters to be set expressed as string, and the distribution identifiers for users and groups are exported.  
See [Data type formatting](#) on page 354.
- The double quote character, ", is not allowed in the matrix definition for export to CSV.
- Comparison operators are translated to string values.  
See [Comparison operators in CSV files](#) on page 354.

### Importing matrix definitions from CSV

- 1 Open the matrix details and click the **Definition** tab.
- 2 Click **Import from CSV** and select a CSV file to import.

The imported matrix overwrites the existing matrix rows.

- 3 Click **Save** to confirm the new matrix definition.

**Note:**

- An import is not performed if there are errors in the import file. These validations are performed upon import:
  - The column names in the CSV file must match the matrix definition.
  - The values in the CSV file must match the data type of the column.
  - User and group identifiers from the import file must be valid identifiers from IFS. See [Data type formatting](#) on page 354.
  - Parameter placeholders must match existing column parameters and data type.
  - The double quote character, “, is not allowed in the import CSV file.
- Up to 500 validation errors are displayed upon import.

## Comparison operators in CSV files

This table shows the comparison operators used in the CSV files:

Operator in matrix definition	Operator in CSV file
=	EQUALS
>=	>EQUALS
<=	<EQUALS

**Note:** Other operators are exported with the same syntax as in the matrix definition.

## Data type formatting

This table shows how data types are formatted:

Data type	Formatted as string in CSV	Condition example CSV
STRING	The string as is. Double quotes are not allowed.	EQUALS This is a string
INTEGER	Number as string.	<EQUALS 100
DECIMAL	Number as represented in the UI. The browser locale must match the system locale where the CSV file was created.	>EQUALS 3.14
BOOLEAN	TRUE or FALSE	EQUALS TRUE

Data type	Formatted as string in CSV	Condition example CSV
DATETIME (UTC)	Time that is converted to UTC. Expressed in this format: yyyy-MM-dd HH:mm:ss	< 2017-06-10 12:24:33
DATE	Formatted as yyyy-MM-dd	EQUALS 2017-06-08
HYPERLINK	The link as string.	EQUALS https://www.infor.com
USER	The user identifier as string. If Restriction from a specific group is configured, the data type is represented as USER(Group name).	EQUALS someone@infor.com
GROUP	The group name as string.	<> TestGroup
CODE	The code as string. The data type is represented as CODE(Codelist name).	EQUALS Approved
Distribution to User	Formatted as USER_FROM_IFS:<user identifier> where <user identifier> is the <b>IFS Person ID</b> field of this user.	USER_FROM_IFS:1001
Distribution to Group	Formatted as GROUP_FROM_IFS:<group name> where <group name> is the name of the distribution group.	GROUP_FROM_IFS:IONQA

## Viewing matrices

### Filters

On the **Matrices** page, a list of tiles is displayed for all matrices in the system. You can use the filter toolbar to filter the list. Use these options:

- Turn on or turn off the switches to show or hide the active and inactive matrices. By default, both switches are on. Therefore, all matrices in the system are displayed.
- Select a sub status from the drop-down filter. This applies to the list of matrices resulting for the switch selections. For example: all matrices or only active matrices.

This table shows the sub statuses you can chose:

Status	Description
All	Show all matrices, with all statuses.
Approved	Show matrices that have been approved and are Active, and that have not been modified since the activation.
Draft	Show these matrices: <ul style="list-style-type: none"> <li>• Matrices that are Inactive.</li> <li>• Matrices that are Active but have a newer version in Draft, which has not yet been approved.</li> </ul>
Pending Approval	Show matrices that are submitted for approval, but are not approved yet. These matrices can have these statuses: <ul style="list-style-type: none"> <li>• Inactive Pending Approval if the matrix was never approved before</li> <li>• Active Pending Approval</li> </ul>
Rejected	Show matrices that have been submitted for approval and have been rejected. These matrices can have these statuses: <ul style="list-style-type: none"> <li>• Inactive Rejected if the matrix was never approved before</li> <li>• Active Rejected if a previous version of the matrix has been approved before.</li> </ul>

- On the list of matrices that results from the status filtering, you can use a textual filter. The search text is applied to the matrix name, description, or author (**Last Updated by**) properties. All matrices that contain this string in one of these fields are displayed.

## Viewing previous matrix versions

You can view previous matrix versions as read-only information.

- 1 Go to the details view of a matrix.
- 2 Select one version from the **Versions** widget in the left panel.

The details view is updated to show this information:

- Description
- Last update by information
- Definition of that version

## Approving and activating a matrix

For a matrix with the Pending Approval status:

- 1 Open the matrix detail view.
- 2 Click **Approve**.
- 3 Check whether the matrix status has become Active.  
This version of the matrix cannot be edited. A new version (Draft) is created as a copy from the current version, and is available for editing.
- 4 To reject a matrix, click **Reject**. Specify a reason for rejection. The current matrix version becomes editable again for the DECISIONSERVICE-Editor. The editor can change the matrix and submit it again for approval.

## Simulating the matrix execution

To simulate the matrix execution, the DECISIONSERVICE-Editor or DECISIONSERVICE-Approver roles are required.

With the Simulation functionality you can verify which approval chain will be built for a set of input values.

To use the simulation functionality:

- 1 Open the details page of the matrix you created.
- 2 From the list of matrices, click the matrix tile to open the details page.
- 3 Click the **Definition** tab.
- 4 Click **SIMULATE**. The switch is now ON.
- 5 Verify whether the matrix view has changed. Each column has an input field that requires a value.
- 6 Specify a value in each field and press **Tab** or **Enter** to move to the next field. When all values are specified and no field is marked red, the simulation is executed.  
The rows for which all conditions evaluate to True for the specified values, are displayed.  
The distribution column, shows how the approval chain will be built. Approvers will receive a task in the order of the rows, from top to bottom.
- 7 To exit the simulation mode, click **SIMULATE**. The switch is now OFF.  
The whole matrix is now displayed, as it was defined.

## Using an approval matrix from Workflow

You can use an active approval matrix in Workflow in the configuration of a task chain step. At runtime, this is how the workflow approval chain is executed:

- The matrix is evaluated based on the input values specified in the parameter mapping. The first approver is determined.

- If no row has evaluated to True, the matrix evaluation is skipped. The workflow moves to the next step. If one row is evaluated, an approval task is sent to the first approver.
- If the user approves, the matrix rules are evaluated again. The next approver is determined. An approval task is generated for the next approver.
- After all users in the chain approved, the approval chain stops. The result is “Approved”.
- If one user rejected, the approval chain stops immediately. The result is “Rejected”.

**Note:** You can specify other labels for the “Approve” and “Reject” user actions.

To configure a task chain step with shared approval matrix:

- 1 In the workflow model, select the **Task Chain** step.
- 2 Click the **Approval Matrix** tab.
- 3 Select **Use Business Rules**.
- 4 Select an approval matrix name from the list of active matrices.
- 5 The matrix input parameters of the selected matrix are displayed. Map the workflow parameters to the matrix parameters.
- 6 Save and activate the workflow definition.

## Using a decision matrix from Workflow

You can use an active decision matrix in Workflow in the configuration of a decision table step. At runtime, the decision matrix is evaluated each time the decision table step is executed.

The decision matrix can be updated and approved when the workflow is running. The next execution of the decision table step uses the latest version of the matrix. It is not required to re-activate the workflow to use the latest matrix version.

When a decision matrix is evaluated, the rows from the **Conditions** area are evaluated one by one, from top to bottom. When a first row is found that evaluates to true, the values from the **Actions** columns are returned for the parameters to be set. The matrix evaluation stops.

## Configuring decision table step with shared decision matrix

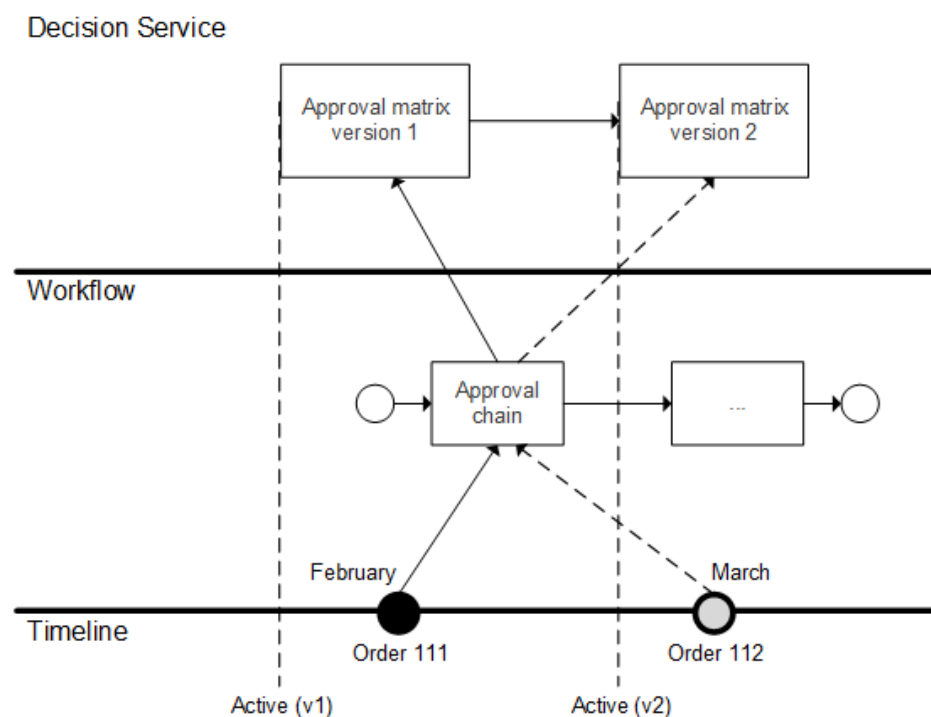
- 1 In the workflow model, select the Decision Table step.
- 2 In the **Decision Table Properties** pane, select the option **UseBusiness Rules**.
- 3 Select a decision matrix name from the list of active matrices.  
The parameters of the selected matrix are displayed and there are two columns for mapping workflow parameters, input and output.
- 4 Map all input parameters to workflow parameters.
- 5 Map output parameters to workflow parameters or ignore them.
- 6 At runtime, the workflow parameters mapped as input are evaluated in the matrix conditions.  
When a match is found, the workflow parameters mapped as output contain the values set from the matrix. When there is no match in the matrix rules, the workflow parameter values remain unchanged.

## Workflow and matrix versions

When an approval chain is started to evaluate a matrix, the same matrix version is used until the approval process triggered by this chain ends. See "Order 111" and the corresponding continuous arrows in the diagram.

Each time the workflow is triggered by a new document, an evaluation of the latest active version of a matrix is started. See "Order 112" and the corresponding dashed arrows in the diagram.

This diagram shows how matrix versions are used in an approval chain:



When a decision matrix is used in a workflow, the latest active Matrix version is used each time the workflow decision table step is executed.

## Page overviews

### Matrices

#### Matrices

On this page you can:

- See the overview of all matrices and their status.

- Filter the list by matrix name, description, or author name.
- Filter the list by matrix status, type, or activation status.
- Click **ABOUT** to open the **About** dialog box that shows the application build number.
- Import or export matrix definitions and create new matrices.

To perform these actions, the DECISIONSERVICE-Editor role is required.

## Approval Matrix

### Matrices

Open the **Approval Matrix** detail page.

On this page you can:

- See the matrix definition and its previous versions.
- Change the matrix. For example, to make edits or to approve and activate the matrix.
- Import/Export the matrix to a `CSV` file.
- Simulate the matrix execution.

To perform these actions, the DECISIONSERVICE-Editor or DECISIONSERVICE-Approver security role is required.

## Decision Matrix

### Matrices

Open the **Decision Matrix** detail page.

On this page you can:

- See the matrix definition and its previous versions.
- Change the matrix. For example, to make edits or to approve and activate the matrix.
- Import/Export the matrix to a `CSV` file.
- Simulate the matrix execution.

To perform these actions, the DECISIONSERVICE-Editor or DECISIONSERVICE-Approver security role is required.



## Chapter 13: ION OneView, management and troubleshooting

In ION OneView you can track the complete process for ION processed documents from a single place. Message Trace that was available until ION 10.3.x is replaced with ION OneView.

### Management

The Manage section acts as a control center for ION. The views of this section represent the ION Service. It is used for monitoring, troubleshooting and solving errors.

### Troubleshooting

The Troubleshooting information describes how to diagnose and solve issues within the application.

## ION OneView

You can find ION OneView in the **OneView** menu in ION Desk.



The main capabilities of ION OneView include:

- Track Business Documents from a single consolidated view; Search for documents using different search criteria.
- View all ION Components that were triggered by the incoming document. This includes Connection Points, Document Flow filters and Content-Based Routing, Mappings, ION Engines (Monitor, Workflow, Pulse...), Monitors, Activation Policies and Workflows.
- View more details of these ION Components:
  - Detailed properties for each ION Component displayed.
  - List of events that were logged by each ION Component when processing the message.
  - Drill-down views to display the appropriate manage pages that are related to the selected ION Component.
- Visibility & Correlation of the different messages sent or generated:
  - Original messages that triggered the whole list of components displayed.
  - Confirm BODs messages that were generated due to any error when processing the original message.
  - Mapped/Updated messages that were created during the processing of the original message.

- View content of messages. This is only available for authorized users - based on the roles and permissions configured. When viewing the contents of a message you can format the message to make it more readable. By default the message contents are displayed as is. When the used format in the message content is not readable, download the whole message and determine its content with your tools.

See Document Authorizations in [ION Desk authorizations](#) on page 500.

This table shows the message content window toolbar:

Name	Type	Description
Download		Click to download a file to a local folder with this name: <MessageID>.<extension> The extension is omitted for ANY document types.
Format		Click to format the message content. Click again to restore the original content.

ION OneView provides these different views:




- Timeline View:** This is a graphical representation of the message trip within ION and all ION Components that were triggered during that trip.
- Advanced View:** This is a list of all the events that were logged for the message during its processing within ION.






The functionalities and capabilities of ION OneView are explained for both views in other sections.

## Search documents in ION OneView

Open the menu and navigate to **OneView**. The search panel with basic search filters is opened.

This table shows the controls used on the **OneView** page toolbar:




Name	Type	Description
Refresh		Refreshes the live data. Some of the search fields drop-down values are dynamically generated based on the available data to search on.
Reset		Clears up custom filters and resets all the filters to default.
Filters		Saves the search criteria. If you save the search criteria, the <b>Filters</b> icon becomes 'filled'. See the next icon in this table.

Name	Type	Description
Filters		<p>Indicates that the search criteria are saved and are used if you bookmark the page.</p> <p>To bookmark the page, click the <b>Bookmarks</b> icon in the Infor Ming.le top navigation panel.</p> <p>If you open the saved bookmark from the Infor Ming.le <b>Bookmarks</b> menu, the <b>OneView</b> page opens and the saved search criteria are applied automatically.</p> <p>If you edit the saved search criteria, the icon becomes "unfilled". To save the updated search criteria and use them when bookmarking the page, click the <b>Filters</b> icon again.</p>
Group by	drop-down	Selects a custom filter to use for grouping of the search results together with the document type and document name. Available options are Document ID (default), Sender or Date.
Timeline		Selects Timeline view for the search results. The icon is faded when Timeline view is selected.
Advanced		Selects Advanced view for the search results. The icon is faded when Advanced view is selected.
Search		Opens or closes the search panel. Use to gain more space for the results.
Settings		Opens or closes the settings panel. Use to customize the view of the components by show or hide components. These settings are saved for each ION user logged. The settings are saved within a specific machine.

When available, you can use the **Text field filter** to get more precise results and search faster.

Click the **SEARCH** text button to start searching with the specified filters.


This table shows the text filter options:

Name	Type	Description
Starts with		Shows the documents where the value starts with the search string.
Equals		Shows the documents where the value is exactly identical to the search string.
Contains		Shows all the documents where the value contains the search string.

## Basic search

You can use a basic search with most commonly used filters.

This table shows the basic search fields:

Search Criteria	Description
Date and Time Range	<p>Represents the date-time in which messages started their processing within ION (<b>Message entered ION</b> event).</p> <p>Click this button to select the search period:</p>  <ul style="list-style-type: none"> <li>• <b>Last hours (1-23)</b>: selects documents from date range starting from exactly (selected value from 1 to 23) multiplied by 60 minutes in the past till current date-time in UTC.</li> <li>• <b>Last 24 hours</b>: selects documents from date range starting from exactly 24 hours in the past till current date-time in UTC.</li> <li>• <b>Last 7 days</b>: selects documents from date range starting from exactly seven days in the past till current date-time in UTC.</li> <li>• <b>Date Time Range</b>: selects documents with date range between the specified "Start Date" and "End Date" including time (hh.mm). "End Date-Time" must be later than "Start Date-Time".</li> <li>• <b>All</b>: selects anything until the current date-time.</li> </ul>
Document Name	<p>Allows to search by a document name, for example <code>Sync.SalesOrder</code>. Select documents from the drop-down list. The list shows the documents that are processed by ION and for which the current user has view permission. Click <b>Refresh</b> to update that list. To filter the list of available documents use a search field on top of the list. Specify a search string in the field to see only those documents that contain the specified string.</p> <p><b>Note:</b> This filter is only working when the selected Display View is "Timeline".</p>
Document ID from message content	<p>Specify one or more Document IDs, comma-separated. Use the text filter to limit the search results and improve the response time.</p> <p>If the document ID is not found in the message content. The document ID from the message header is used, unless it is blank or wrapped in double quotes.</p> <p><b>Note:</b> ION OneView engine does not support field values enclosed in double quotes. The double quotes indicate that the search value is a "phrase" meaning the search engine searches for an exact match. The use of values that are enclosed in double quotes is forbidden.</p>

## Advanced search

You can search by important fields from document contents, using values of required and optional header fields. You can also search for documents that are handled in certain objects such as connection point, activation policy, and mapping.

Button	Description
+	With this button you can add an advanced search filter.
×	Use this "x" to remove any filter. You can find this "x" above each field.

The **Field Type** drop-down has a list of search fields grouped by categories. The options available depend on the actual search data. Click **Refresh** to update the lists.

When searching you can use any combination of elements from these groups, combined with the three basic criteria.

After selecting the field type, specify a value in the matching field. When selected, the value is removed from the list of options.

## Export Advanced search results

You can export the search results as a CSV-formatted file by clicking the **Export** icon. A maximum of 1000 pages can be exported. Such an export process can take some time. To save time, you can specify a lower number of pages.

The number of pages is counted as: 1 Page = 25 events. A maximum of 1000 pages means 25000 events.

## Standard fields

To search by values of mandatory header fields, select a field under the **STANDARD FIELDS** category in the **Field Type** drop-down.

This table shows the **STANDARD FIELDS** category:

Message ID	Specify one or more message IDs, comma-separated in the text field. You can specify only a part of the Message ID and ION OneView will search for any matching documents that contain the string specified in their Message ID. Use the text filter to limit the search results.
Sender Logical ID	Select one or more Logical IDs from the available drop-down list. This list reflects the Sent From Logical IDs for documents that are currently processed by ION for documents the user has view permission. Click <b>Refresh</b> to update the list. To filter the list of logical identifiers, specify a search string in the drop-down. Only the logical identifiers that start with the specified string display. Select or clear identifiers to add or remove them from the search filter.

Tenant	Select one or more tenants from the available drop-down list. This list reflects the tenants for documents that are currently processed by ION and for which the user has view permissions. Click <b>Refresh</b> to update the list.  To filter the list of tenants, specify a search string in the <b>Tenant</b> drop-down. Only the Tenants that start with the specified string are displayed. Select or clear tenant names to add or remove them from the search filter.
--------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Message processing

In ION OneView, you can search on the status of a message.

For example, whether the message is still in progress or resulted in an error or warning. Additionally, the status is displayed in the search result and the timeline. Select a field under the MESSAGE PROCESSING category in the **Field Type** drop-down.

These options are available:




- **Processing Status** - Select this option to search for the messages that are in progress, that is, in flight, or completed.
- **Processing Exception** - Select this option to search for the messages that resulted in errors or warnings, that were not routed or were filtered out.



This table shows the MESSAGE PROCESSING category:

Processing Status	Select one or multiple options from the available list: <ul style="list-style-type: none"> <li>• <b>In Flight</b></li> <li>• <b>Completed</b></li> </ul>
Processing Exception	Select one or multiple options from the available list: <ul style="list-style-type: none"> <li>• <b>Error</b></li> <li>• <b>Warning</b></li> <li>• <b>Filtered Out</b></li> <li>• <b>No Route</b></li> </ul>

The status indicators are displayed in the search results in the left panel or in the timeline.

This table shows the status indicators for each category and the locations where they are displayed:

Category	Status	Icon	Usage
Processing Exceptions	Error		In list of messages and timeline
Processing Exceptions	Warning		In list of messages and timeline
Processing Exceptions	No Route		Only in list of messages

Category	Status	Icon	Usage
Processing Exceptions	Filtered out		Only in timeline
Processing Status	In Flight		Only in list of messages
Processing Status	Completed	None	N/a

When you are navigating across the page by selecting other messages or bulbs, the status updates are checked and the status indicators are updated accordingly.

## Message header fields

To search by values of optional header fields, select a field under the **MESSAGE HEADER FIELDS** category in the **Field Type** drop-down list.

This table shows the **Message Header Fields** category:

Accounting Entity (message header)	Specify one or more accounting entities, comma-separated. You can specify only a part of the accounting entity. ION OneView searches for any matching documents that contain the specified string in their Accounting Entity. <b>Note:</b> This filter is only available if the selected Display View is "Timeline".
Document ID (message header)	Specify one or more document IDs, comma-separated. You can specify only a part of the ID. ION OneView searches for any matching documents that contain the specified string in their document ID. <b>Note:</b> This filter is only available if the selected Display View is "Timeline".
Location (message header)	Specify one or more locations, comma-separated in the text field. You can specify only a part of the location. ION OneView searches for any matching documents that contain the specified string in their location from the message header. Use the text filter to limit the search results. <b>Note:</b> This filter is only available if the selected Display View is "Timeline".
Source (file name)	Specify one or more file names, comma-separated in the text field. You can specify only a part of the file name. ION OneView searches for any matching Sources (file names) that contain the specified string in their source from the message header. Use the text filter to limit the search results. <b>Note:</b> This filter is only available if the selected Display View is "Timeline".

## Document content fields

To search by important fields from document contents, such as status or description, select a field under the **DOCUMENT CONTENT FIELDS** category in the **Field Type** drop-down.

**Note:** The document content fields are only available when the selected Display View is "Timeline".

This table shows the **Document Content Fields** category:

Accounting Entity (document content)	Specify one or more accounting entities, comma-separated in the text field. You can specify only a part of the accounting entity and ION OneView will search for any matching documents that contain the string specified in their accounting entity. Use the text filter to limit the search results.
Description	Specify description from the message content. You can specify only a part of the description and ION OneView will search for any matching documents that contain the string specified in their description. Use the text filter to limit the search results.
Revision ID	Specify one or more revision IDs, comma-separated in the text field. You can specify only a part of the revision ID and ION OneView will search for any matching documents that contain the string specified in their revision ID. Use the text filter to limit the search results.
Status	Specify Status in the text field. You can specify only a part of the status and ION OneView will search for any matching documents that contain the string specified in their status. Use the text filter to limit the search results.
Variation ID	Specify one or more variation IDs, comma-separated in the text field. You can specify only a part of the variation ID and ION OneView will search for any matching documents that contain the string specified in their variation ID. Use the text filter to limit the search results.

## Retrieved from Data Lake

The same document can be retrieved from Data Lake based on a schedule. The retrieve activity is used in the definition of a data flow. The result is multiple instances of the same document that have different headers but the same content.


You can search for the published documents on a specific date and time that are already retrieved from Data Lake:

- 1 Select the **Retrieved from Data Lake (Date and Time)** check box in the RETRIEVED FROM DATA LAKE category.

Ensure that you have set the display view to Timeline, otherwise the check box is not available.

This table shows the Retrieved from Data Lake category:



Category	Description
Retrieved from Data Lake (Date and Time)	Represents the date and time in which messages were retrieved from data pipelines. To select the search period click the calendar button: 

- 2 Click **OK**
- 3 Select a time period.

## Handled in

You can search for documents by objects that processed, or are processing, the message.

Select an object under the HANDLED IN category in the **Field Type** drop-down list:

- Activation policy
- Alarm Template
- Alarm
- Connection point
- Document Split
- Mapping
- Monitor
- Scripting
- Workflow in document flow

This table shows the advanced search criteria for the “Handled In” category:

Handled in Activation Policy	Select one or more activation policy names from the available drop-down list.
Handled in Alarm Template	Select one or more alarm templates names from the available drop-down list.
Handled in Alarm	Select one or more user names from the available drop-down list of the users who created the alarms on their mobile devices or homepage widget.
Handled in Connection Point	Select one or more connection point names from the available drop-down list.
Handled in Document Split	Select splitter name(s) that are used in document flow.
Handled in Mapping	Select one or more mapping names from the available drop-down list.
Handled in Monitor	Select one or more monitor names from the available drop-down list.
Handled in Workflow in document flow	Select workflow name(s) that are used in document flow.

If the document is not displayed in the search results. The source or destination connection point that handles this document can be in error or paused status. A warning is displayed if one or more active connection points are in error or paused status. Click the hyperlink in the warning message. The **Active Connection Points** page is displayed. On this page, you can find out which connection points are in error or paused status.

### Filtering timeline

When you use "Handled in" to search for documents that are handled in a connection point, activation policy or a workflow. You can filter the timeline view. Only the components that handled a message are shown.

Put the switch above the timeline from **All Components** to **Handled in** to show only the components that handled a message.

Components that precede and follow the component marked with the "Handled" flag are displayed, to keep the context.

### Not handled in

You can also search for documents not handled in a specific mapping or connection point.

For example, to search for a requisition published by LN but not handled in EAM:

- 1 Add the **Handled In** search field.
- 2 Select the **Connection Point** component type.
- 3 In front of the field, switch to **Not**.
- 4 Specify **EAM**.

### Data publishing

Through Data Lake API you can only retrieve the data from Data Lake flows and the data that is added directly to Data Lake without interference of ION.

By default the data from document flows is not visible in the Data Lake, it is stored, in private, for OneView.

When the Data Lake APIs and the JDBC driver must access the data, you can set the private data to public in the Data Lake.

Publish the data in the Timeline View. The date and time range must be restricted. The **Publish to Data Lake** button becomes visible when the advanced search filter **Available in the Data Lake** is used. Search by the value **no** to find the data that is not accessible from the Data Lake.

**Note:** When the data is set to public, the defined Storage Policy is immediately applied to that data.

- 1 Go to **OneView** to search for the „private“ data.
- 2 Select the Timeline view in the page toolbar.
- 3 Click **Add field**.
- 4 Select the **Available in the Data Lake** check box and click **OK**.

- 5 Select **No** as a value in the **Available in the Data Lake** search field.
- 6 Specify a restricted time range.
- 7 Click **Search**.
- 8 Click **Publish to Data Lake**.
- 9 Confirm the publishing.

## Purged from Data Lake

You can purge corrupted data objects from Data Lake either through ION API or from the UI.

Data objects can be purged either in bulks or one by one. If a document is purged, you cannot view or download its content, message payload, in OneView. In that case the Data Lake icon on the timeline is decorated with a bin.

To trace the details of the purge event, select the Data Lake icon decorated with the bin. The purging is on the last component of the message timeline.

## Timeline view search results for documents

Timeline View provides a graphical step-by-step representation of what was processed within ION.

Select the Timeline view option when:

- Searching for a specific type of document.
- You want to know what happened to a certain document, using its Document ID as an identifier.

For example, for Sales Order #123456789 you want to know:

- How many messages were processed regarding that order.
- When was the order placed in ION.
- What happened during the processing of every message that is related to that Sales Order.

Specify the search criteria, Document Name and Document ID, the results are represented in these main steps:

- Document results.
- Messages for selected document.
- Timeline view for selected message.
- Drill Down to Desk pages.

## Document results

When you click **Search**, a list of documents that match the search criteria is displayed in a graphical way.

Each matching document shows this information:

- Document Name. This is the type of documents found, for example; Sales Order.
- Document ID. This is a default grouping filter. You can change this filter to Sender or Date.  
If Document ID is not found in the message content or header, its place stays empty on the document tile
- Number of messages that were/are processed by ION for that document. The colors of the BOD counts change based on how many messages were processed. The exact counts are displayed for numbers less than 1000. For larger counts, the page shows: >1K.

For any information that is displayed in these documents tiles, a tooltip is available to display the full information.

The total number of documents found and the number of messages for these documents are displayed beside the "Documents" title. The results can be sorted by Document Name or ID whether ascending or descending. If results found cannot be displayed in one page, the paging navigation is shown below the documents displayed in the current page.

To refine the search, the search criteria section is still displayed and you can specify the new search criteria and click **Search**. The search results are refreshed and documents that match the new criteria are displayed.

To see the details of a specific document, click the tile of that document. For example to see details about which messages were processed by ION for a specific Sales Order. You can click anything that is displayed in the document tile: Count, Document Name or Document ID. The details section for the selected document, messages & Timeline View for these messages, are displayed underneath the selected document tile.

Details about this section are explained in:

- [Messages for selected document](#) on page 372
- [Timeline for selected message](#) on page 373

The rest of the Documents Results are still displayed above and below the details section. In case you select another document to view its details, it should be possible without navigating back and forth. In addition, if you do not longer want to see the documents details, click the selected document tile to close the details section.

## Messages for selected document

For a selected document, you can see all messages that were/are processed by ION. The messages are sorted ascending, by default, and by the date-time at which their processing started. This messages list is displayed on the left-side of the documents details section. Information that is displayed in this list includes:

- **Document Name:** This is represented by both the Verb and Document Type of the selected message.
- **Count Instances:** This is a number of unique records in the payload. When a publishing application has not provided a count. Click the hyperlink to request the count and see the number in the UI.
- **Source Name:** This is the Logical ID of the source application/component that created the message. This is the ION Component that logged the **Message entered ION** event.

- **Date-Time:** This represents the date-time in UTC of the **Message entered ION** event for the selected message.

For any information that is displayed in this messages list, if it is too long, a **tooltip** is available to display the full information.

The list of messages can be sorted by Document Name, Source Name or Date-Time whether ascending or descending. If many messages are available, a scroll bar displays to navigate through the list.

You can view the timeline of a certain message. Scroll to that message and select its entry in the list. Its timeline is displayed on the right-side of the document details section. By default, the timeline of the first message that is listed is displayed.


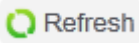






**Note:** A maximum of 1000 messages for a document can be shown in **OneView Timeline**. Refine the date-time range criteria to shorten the list of messages found for a document.

## Timeline for selected message

Timeline View provides a graphical representation for the processing of messages within ION.

It displays all the components (applications) that were used / triggered by the selected message.

Timeline View shows the processing start and end time for the message and which components were involved.

Icon	Description
	Event Details icon
	Refresh icon
	Original document icon
	Confirm BOD icon
	Transformed/Mapped document icon
	Zoom out to make time line smaller and farther away
	Zoom in to make timeline bigger and closer.
	Use to reset the timeline view to 100%.

Timeline View provides these capabilities:

- You can view the components triggered by the message within ION.
- Click the component icon to view the summary information about each triggered component. The component properties and the events logged by this component are shown. The selected component is highlighted and an Event Details icon is available. Click the Event Details icon to show all events details.

- You can view summary information about each message handled (original, mapped, updated and Confirm BOD). Selected message is highlighted.
- You can view the content of the messages that are handled during the timeline. Use this window to copy content to the clipboard, see [Message Content Window](#) on page 374.
- You can drill-down to the details of the selected component to see configuration and runtime information. Use the available hyperlinks in the components properties section.  
Details to drill-down are explained in "Drill down to Desk pages".
- Right-click the message icon to navigate to other messages timeline with timeline link for original, mapped, updated and Confirm BOD messages.
- Click the Refresh icon to manually reload the Timeline.
- A Legend is available, showing the name of every component displayed. Right-clicking the component icons shows/hides those components from the timeline, with the exception of the document icons.
- Click the Splitter component, right-click the message icon with a counter. A window is displayed showing the sequence numbers of instances after the split. Use this window to see the timeline of the selected instance from the split collection instead of the original timeline. You can click **Refresh** to return to the original timeline.

## Message Content Window

On the message timeline, double-click the document icon to open the message (document) content window. The window shows payload of the original, mapped, updated or Confirm BOD messages.

These buttons are available at the top of the window:

- Format
- Download
- Close

This table describes the icons in the header of the message content window:

Name	Description
Format	Use this button to apply formatting to the original content for easy reading. You can restore the original content by clicking the icon another time. When the content is formatted, the icon color is blue. Notes: The original content is shown by default. The original content can look formatted. The button is hidden if formatting is not applicable.
Download	Use this button to download the original content to view it locally. This is helpful when the content is available but cannot be displayed for any reason, such as size or binary format. Notes: The original content is downloaded even if the payload was formatted by a user. The button is hidden when the document is not retrieved, was empty or purged.
Close	Use this button to close the window.

The information available at the bottom of the window:

- Document Name
- File Type\*
- Message Size\*\*
- Counter of instances

\*The file type displayed in the message content pop-up window matches the current format type in Data Catalog, not the type of the data at the time of publishing. If you update the data format in the settings of AnySQL Connection Point in ION Desk and ION generates metadata, the new file type is used in the message content window for the already published data.

\*\*Only one size is available for the payload content. The label clarifies, which size was used – compressed or original.

This table shows the formats supported in OneView:

Message Format	OneView can display content	Details
XML (BODs)	yes	The message content is validated by OneView and indicates, if the content is invalid XML.
JSON (Conventional)	yes	The message content is validated by OneView and indicates, if the content is invalid JSON.
JSON (Newline-delimited) – NDJSON	yes	The message content is validated by OneView and indicates, if the content is invalid NDJSON.
DSV (CSV, TSV, PSV, Other) - delimited data formats	yes	DSV formats are displayed as plain text, no highlighting, no formatting. No content validation.
ANY	no	The file formats using the Data Catalog's Any data format. No content validation.

Infor ION Data Catalog recommends that you use one of these syntaxes:

- `$.*['propertyName']` syntax (for conventional JSON objects)
- `$['propertyName']` (for newline-delimited JSON)







## Drill down to Desk pages

In ION OneView details are provided that relate to the documents that were processed in ION.






A summary of the ION components that were involved in this processing is shown. Clicking on a component icon in the timeline view shows a list of applicable properties for that component. Some of







these properties are displayed as hyperlinks. You can see more details for the viewed ION components by drilling down to other ION Desk pages. Mainly Manage pages.




This table lists the places where the drill-down option is available and which page is displayed:

ION Component	Hyperlinks Properties	ION Desk Page
Connection Point 	Usage: Document flow in which the connection point is or was used.	Active Data Flow – details page
	Name Connection Point Name	Active Connection Point - details page
ION API 	Usage: Document Flow Name in which the API activity is or was defined.	Active Connection Point - details page
	Name: Connection Point Name	
Filter Flow 	Usage: Document Flow in which the filter was defined.	Active Data Flows - details page
Content-Based Routing Flow 	Usage: Document Flow Name in which the Content-based Routing is or was defined.	Active Data Flows - details page
Splitter 	Usage: Document Flow Name in which the splitter is or was used.	Active Data Flows - details page
Mapping 	Usage: Document Flow Name in which the mapping is or was used.	Active Data Flows - details page
	Name: Mapping Name Mapping Version #	Mapper



ION Component	Hyperlinks Properties	ION Desk Page
Workflow from Document Flow 	Usage: Document Flow Name in which the workflow activity is or was defined <hr/> Workflow ID: Workflow Instance Id # <hr/> Name Used: Workflow Name <hr/> Name: Activity Name	Active Data Flows - details page <hr/> Active / Archive Work Flows - list page - row for Instance # <hr/> n/a <hr/> n/a
Data Lake 	Document Flow in which the connection point is or was used.	Active Data Flows – details page
Activation Policy / Monitor Engine 	Activation Policy Name: Activation Policy Instance Name <hr/> Monitor Name: Monitor Instance Name <hr/> Workflow ID: Workflow Instance IDs #	Active/Archived Activation Policies- list page - row for Instance name major version <hr/> Active/Archived Monitors - list page - row for Instance name major version <hr/> Active/Archived Workflow Instances - list page – row for Instance #
Pulse/Activities Engine 	Alert ID: Pulse Item Id #	Active / Archive Activities - list page – row for Instance #
Activation Policy 	Activation Policy Name: Activation Policy Name <hr/> Workflow Name: Work flow Name <hr/> Workflow ID Workflow Instance IDs # <hr/> Alert ID: Pulse Item ID#	Active / Archived Activation Policies - list page - row for name and major version. <hr/> Active / Archive Workflows – list page - row for Instance # <hr/> Active/Archived Workflow Instance - details page <hr/> Active/Activity Instance - details page (details for Alert #)





ION Component	Hyperlinks Properties	ION Desk Page
Workflow 	Workflow Name: Workflow Name <hr/> Workflow ID: Workflow Instance ID # <hr/> Alert ID: Pulse Item ID #	Active / Archive Workflows - list page - row for instance # <hr/> Active / Archive Workflow Instance - details page <hr/> Active Activity instance details page - details for Alert #
Monitor 	Monitor Version: Monitor Instance Name Monitor Version Monitor Status <hr/> Alert ID Alert ID #	Active / Archive Monitors - list page - row for Instance name major version <hr/> Active/Activity instance - details page - details for Alert #
Alarm Template 	Alarm Template Name Alarm Template Version # <hr/> Alert ID Alert IDs #	Active / Archive Alarm Templates - list page - row for Instance name major version <hr/> Active Activity instance - details page - details for Alert #
Original Document 	Double-click the icon	A window shows the message content.
Updated Document / Mapped Document 	Click the icon	Left-click the transformed document icon to open a window with the time line for the new message id. You can also move the window and see both time lines next to each other. To return to the original time line, close the window. Drill-downs from the floating window are not supported. Alternatively, right-click the transformed document icon to load the time line of the transformed document. To return to the original time line, click <b>Refresh</b> .
Confirm BOD 	Message ID: Message #	Error BODs - details pages

ION Component	Hyperlinks Properties	ION Desk Page
Manual Action 	n/a	n/a
Unknown Component / Event Source 	n/a	n/a
Workflow Engine 	n/a	n/a
Confirm BOD Manager Engine 	n/a	n/a

## Timeline indicators for alerts

You can find the graphical indicators directly on the timeline, next to the icons of monitors, activation policies, or alarms. The alert indicators are not displayed on the message tile.

This table shows the alert indicators and their description:

Alert indicator	Description
	The "yellow triangle" icon shows alert data for monitor, activation policy, and alarm. This icon is not displayed without a counter. "0" indicates "document was evaluated but no alert was generated".
	The "document ignored" icon shows alert data for monitor and activation policy. There is only one use case: document was ignored.
	The "green circle with a white tick" icon indicates that the activation policy has started or canceled an instance of the workflow.
	The "red circle with an exclamation mark" icon indicates that the activation policy failed to start or to cancel a workflow instance.

## ION OneView components / event sources

ION OneView provides information related to the messages that were processed and which components were involved. This information is represented in both views: Timeline and Advanced.

In Timeline View, graphical representation of this information is available. A number of icons are used to display this information. Icons displayed include the event sources which can be configurable components (such as connection points, monitor instances, workflow instance or activation policies) or engines (such as CBM, Activation Policy, Workflow or Pulse) and the messages handled (original, mapped, updated, and Confirm BOD).

Category	Icon	Component / Event Source
Connection Points		Application Connection Point
Connection Points		API Connection Point
Connection Points		Infor Data Lake
Connection Points		Published to Data Lake
Connection Points		Purged from Data Lake
Connection Points		Retrieved from Data Lake (document-based)
Flows		Filter Flow
Flow		Content-Based Routing Flow
Other Activities		Splitter
Other Activities		Mapping
Other Activities		Scripting
Other Activities		Workflow from Document Flow
ION Engines		Activation Policy / Monitor Engine

Category	Icon	Component / Event Source
ION Engines		Workflow Engine
ION Engines		Confirm BOD Manager Engine
ION Engines		Activities Engine
Event Management		Activation Policy
Workflow		Workflow
Event Management		Monitor
Event Management		Alarm Template
Manual Actions		Manual Action
Others		Unknown Component / Event Source
Messages		Original Document / Updated Document / Mapped Document
Messages		Confirm BOD When this icon is shown in the timeline of another document, right-click to navigate to the timeline of the Confirm BOD
Messages		Original Document / Updated Document / Mapped Document with timeline link. Right-click to navigate to the timeline of this document

Components that did not finish the processing of a message can temporarily show the retrying flag:



## ION OneView event types

During the processing of messages within ION, events are logged by ION components handling the messages. All events have these properties by default:

- Event Type
- Event Description
- Creation Date-Time (UTC)

The table lists the currently logged events:

Event type	Description
Message entered ION	This event is logged at the first moment the message is getting into ION through one of its components (such as Connection Points, Engines). This event represents the starting point for the message processing within ION. This event is logged also when an initially marked message as "no route" is resent from the Active Connection Points management page.
Message sent to any connector or engine	This event is logged when a message is delivered to ION component(s) internal queue(s) that further processes this message.
Message consumed by any connector or engine	This event is logged when a message is delivered from the ION component internal queue to be processed by the receiver.
Message delivered by any connector or engine	This event is logged when a message is successfully processed by an ION component and the message is then delivered to the expected external receiver.
Document Updated	<p>This event is logged when a message is updated by an ION component. This is mainly logged when using Database or Webservice Connection Points as intermediate steps and when using Workflow Activity in a document flow.</p> <p><b>Note:</b> The original message that was sent to these connection points are not updated. A new message is created, based on that original document together with the updates done by the ION component. The new message is visible from ION OneView and its timeline is accessible as well.</p>
Message migrated to Infor Data Lake	<p>This event is logged when a message is migrated into the Infor Data Lake.</p> <p><b>Note:</b> Migrated documents are subjected to the default Archive Policy when migrated to Infor Data Lake. The time when data must be archived starts from the time when data arrives in the Data Lake.</p> <p>If the archive policy is to move the documents to Infrequent Access in 30 days. The data is migrated to Data Lake first and is archived after 30 days.</p>
Document triggered	This event is similar to "Document updated" or "Document mapped". It is logged for all trigger scenarios containing the original message id.
Message cannot be routed	This event is logged when the destination that is specified in the message ToLogicalID cannot be found. In this case, ION cannot send this message to any destination.

Event type	Description
Confirm BOD generated	This event is logged when processing a message by one of ION components and an error occurs that causes the generation of a Confirm BOD. The new Confirm BOD message is visible from ION OneView and its timeline is accessible as well.
Confirm BOD stored	This event is logged when a Confirm BOD that was sent by an application or was created by one of ION components during processing a BOD, is stored in the Confirm BOD Engine.
Message resubmitted	This event is logged when an ION user resubmits a Confirm BOD from <b>Connect &gt; Error BODs</b> page. The event is shown on the original time line of the source message id. The original message processing statuses are kept for tracing purposes."
Confirm BOD marked as handled	This event is logged when an ION user resubmits a Confirm BOD from the <b>Connect &gt; Error BODs</b> page and it is automatically sent to Handled. Or when an ION user selects a Confirm BOD from the <b>Connect &gt; Error BODs</b> page and manually marks it as Handled.
Confirm BOD marked as unhandled	This event is logged when an ION user selects a Confirm BOD from <b>Connect &gt; Error BODs</b> page and manually marks it as Unhandled.
Delivery Failed, retry started	This event is logged when an error occurs during the delivery of a message. For example, the database is not accessible. Then the ION component starts retrying to deliver the message.
Document does not match filter	This event is logged when an incoming message does not match any of the conditions configured in a Filter or a Content-Based Routing flow. In this case, the message is dropped.
Document matches filter	This event is logged when an incoming message matches a condition configured in a Filter or a Content-Based Routing flow.
Document mapped	This event is logged when a message is mapped by an ION component. This is mainly logged when using mapping activities in a document flow. <b>Note:</b> The original message that was sent to these mapping activities are not changed. A new message is created, based on the logic of the configured mapping. The new message is visible from ION OneView and its timeline is accessible.
Document Flow Triggered Workflow	This event is logged when a message is used to start a workflow from document flow. If the workflow started successfully, the workflow instance ID is available. <b>Note:</b> The original message that was used to start the workflow is not updated. A new message is created, that is based on that original document, together with the updates that are done by the workflow. The new message is visible from ION OneView and its timeline is accessible.
Monitor Instance Triggered	This event is logged for every monitor that is interested in the document name of the incoming message. The monitor rule is evaluated based on the incoming message. If the rules evaluate to true, then an alert is created.

Event type	Description
Alarm Template Triggered	This event is logged for every alarm template that was used in the alarm instance to evaluate the incoming document.
Activation Policy Instance Triggered	This event is logged for every activation policy that is interested in the document name of the incoming message. The activation policy rule is evaluated based on the incoming message. If the rules evaluate to true, a workflow is started or cancelled based on the activation policy type. An "Activation Policy Started" event is logged. If the rules do not evaluate to true, then no other events are logged.
Activation Policy started Workflow	This event is logged for every Activation Policy that is interested in the document name of the incoming message and its rules evaluation is true. This is based on the incoming message and a workflow instance was successfully created.
Activation Policy failed to start Workflow	This event is logged for every Activation Policy that is interested in the document name of the incoming message and its rules evaluation is true. This is based on the incoming message but a workflow instance failed to get created for some reason. For example: a workflow instance for the same document id already exists
Activation Policy Instance Completed	This event is logged for every Activation Policy that is monitoring the document name of the incoming message. The activation policy triggers the start/cancellation of a workflow. When this workflow is completed / cancelled successfully, this event is logged.
Workflow Instance Created	This event is logged when the message causes the creation of a workflow instance. This is mainly done when a <code>ProcessWorkflow</code> message is processed that starts a workflow.
Workflow Instance Cancelled	This event is logged when the message causes the cancellation of a workflow instance. This is mainly done when a <code>ProcessWorkflow</code> message is processed that cancels a workflow.
Workflow Instance Failed	This event is logged when the message ( <code>ProcessWorkflow</code> ) causes the creation of a workflow instance and the execution of this workflow fails.
Workflow Instance Completed	This event is logged when the message ( <code>ProcessWorkflow</code> ) causes the creation of a workflow instance and the execution of this workflow is finished.
Pulse Item Created	This event is logged when the message causes the creation of a pulse item (Alert, Task and Notification). This is mainly done when a <code>ProcessPulseAlert</code> or <code>PulseNotification</code> or <code>ProcessPulseTask</code> message is processed by ION and a pulse item is created.
Pulse Item Changed	This event is logged when the message causes the creation of a pulse item (Alert, Task and Notification) and then this Pulse Item changed. For every change, the event is logged. This also happens if the message causes the cancellation of a Pulse Item. This is mainly done when a <code>ProcessPulseAlert</code> or <code>PulseNotification</code> or <code>ProcessPulseTask</code> message is processed by ION and a pulse item is cancelled.



Event type	Description
Document was split	This event is logged when a document was split into several instances. This is mainly logged when the splitter was used in a document flow.
Retrieved from Data Lake	This event is logged when a scheduled retrieval from Data Lake occurs of an instance of the same data object.
Purged from Data Lake	This event is logged when a document was successfully purged from Data Lake.
Document Processed by script	This event is logged when a document was processed by a script from one document to another document.

## Advanced view search results

Advanced View represents a list of what was processed within ION.

Users select the Advanced view option in these situations:

- Searching for a specific message.
- Viewing the list of events logged within a date-range for a certain document or message.
- Exporting the results of an advanced search to excel or other reporting tool to process them.

The user specifies the required search criteria such as Event Type or Document ID. The results are represented in one step:

- Messages and Related Events

You can export up to 10 000 records or 400 pages given there are 25 records per 1 page. You can minimize the number of pages during the export or minimize the search results by filtering.

## Messages and related events

When clicking **Search**, a list of messages that match the search criteria is listed together with events that are logged for these messages. The information that is displayed includes:

- Message ID
- Document Type
- Creation Date-Time
- Document ID
- Accounting Entity
- Location
- Sender Logical ID
- Event Type
- Event Date-Time
- Event Source Type
- Event Source Logical ID

For any information that is displayed in this list a tooltip is available when the list is too long. The full information is displayed.

The total number of BOD-related Events found are displayed between brackets beside the "Messages & Events" title. If the results cannot be displayed in one page, the paging navigation is shown below the list displayed in the current page.

To refine the search, the search criteria section is still displayed and the user can specify the new search criteria and click **Search**. The search results are refreshed and list of messages and events that match the new criteria are displayed.

Click the message icon to see the details of a specific message. A window is displayed showing the content of the message. This window enables the 'Copy' of the content to clipboard.

Click **Event Details** to see the details of a specific event. If clicked, a window is displayed with all events details.

## Manage ION Connect

You can find the options to manage the ION Connect components of ION Service in ION Desk.

Select **Connect** in the left menu.

## Active connection points

Click the hamburger icon to open the menu. Select **Connect > Active Connection Points**.

This page shows the connection points that are active.

For each active connection point these details are displayed:

- Information from the model:
  - **Type**: the type of the connection point (application, database, web service, file or message queue).
  - **Name**: the name of the connection point as specified in the model.
  - **Logical ID**: the logical ID of the connection point, which is derived from the type and the name.
- Status information from the ION Service:
  - **Last Activated On**: the date and time of the last activation of a document flow containing the connection point.
  - **Status**: the current status of the connection point in the ION Service. The status is OK if:
    - The connector is available and, for application connection points, a connection to the application can be established.
    - The polling daemon is running to read messages from the application's outbox.

If one of the criteria is not met, the status is Error. Note that an Error status does not indicate a persistent failure. It can also mean that a connection is temporarily unavailable or the application server is busy and does not respond.

- Pause status of connection point from the ION Service:
  - **Receiving Paused:** Use this option to pause the delivery of messages to the connection point.
  - **Sending Paused:** Use this option to pause the pickup of new messages from the connection point.

For details, see [Pausing message processing of connection points](#) on page 388.

- Statistics from the ION Service on handled and unhandled messages:
  - **Pending (Incoming Pending Messages):** the number of messages that are sent to the connection point, but not yet delivered. They are waiting because no connection can be established, or because the connection point is still busy processing older messages.
  - **Received Unprocessed:** the number of messages that are posted in the in-box of the connection point, but not yet processed. The status in the in-box is 0. This is only relevant for application connection points of type Infor Application and Cloud 2.0, because other types of connection points do not use an in-box. For Cloud 2.0 this indicates the number of messages already sent from the on-premises ION to the Cloud application but not yet processed by the Cloud application.
  - **Received:** the number of messages that are received and processed by the connection point. Note that this counter increases when connector internally processes a message for the next step. Though the actual transaction can still be in progress by the server. If an incoming BOD resulted in a Confirm BOD being published, the incoming BOD is also counted as processed.
  - **Sent Unprocessed:** the number of messages that are sent from a connection point but are not yet processed by ION. This is not relevant for database and web service connection points, because they are based on a schedule. The resulting BODs are immediately handled in the ION Service. For Cloud 2.0 this indicates the number of messages that are sent by the Cloud application but not yet processed by the on-premises ION.
  - **Sent No Route:** the number of messages that are sent from a connection point but not relevant for any other party. No document flow exists for these messages and they are not relevant for Workflow or Event Management either.
  - **Sent (Sent Processed Messages):** the number of messages that were sent and handled in the ION Service. There are several options for these messages. They are:
    - Processed in the final destination(s).
    - Incoming pending messages.
    - Unprocessed messages for another connection point.
    - Incoming pending messages for a mapping, content-based routing or filter.A processed message can result in a Confirm BOD, when the message was incorrect or the processing in a next step of the flow failed.
  - **No Routes in Outbox:** the number of 'no route' messages that still exist in the Outbox. These messages have status 9 and remain in the in-box until they are cleared or resent. This is only relevant for application connection points of type **Infor Application**. Other types of connection points either do not use an outbox or do not keep 'no route' messages in the outbox.
- In the details view of an Application connection point, these categories are displayed:
  - **Sent Processed Messages in Outbox:** the number of sent processed messages that still exist in the Outbox. These messages have status 1 and remain in the outbox until they are cleared. This is only relevant for application connection points of type **Infor Application**. Other types of connection points either do not use an outbox or do not keep processed messages in the outbox.

- **Received Processed Messages in Inbox:** the number of processed messages that still exist in the in-box. These messages have status **1** and remain in the in-box until they are cleared. This is only relevant for application connection points of type **Infor Application**, because other types of connection points do not use an in-box.
- **Note:**
  - The statistics are empty if they are not applicable for the connection point. For example, The **Sent Processed Messages** in Outbox are empty for database and web service connection points. The statistics can also be empty if the connection point has status **Error**, because the ION Service cannot always determine the correct value then.
  - The messages are counted since the (first) activation of the connection point. The counters for messages in in-box and outbox and the incoming pending messages do not hold any historic data. They show the current number of entries that exist in the in-box or outbox.
  - Use the **Active Data Flows** page, to view the message statistics for mappings, content-based routings and filters.

The status and statistics information are automatically refreshed using a configurable time interval from the setting "Automatic counter update".

Refreshing is done per connection point. The data is gathered asynchronously. Therefore not all counters are displayed simultaneously. For some connection points it can take longer than the refresh interval before the counters are displayed. The previous counters are displayed although the new values are not available. The automatic refresh does not update the list of connection points. For example, connection points that are activated or deactivated meanwhile are not shown in the list automatically.

The automatic refresh does not update the list of connection points. For example, connection points that are activated or deactivated meanwhile are not shown in the list automatically.

To refresh the data on demand, click **Refresh**. Results for each of the listed connection point is displayed once they are available from ION Service. The manual refresh action also refreshes the list of connection points.

To display more details for a connection point, select the connection point and click **Details**. Or double-click the connection point.

To display a list of active data flows that use the connection point, select the connection point and click **Usage**.

## Pausing message processing of connection points

If you need to temporarily stop message processing to and from a specific connection point, you can use the **Pause** option. For example, during an application's maintenance window, its inbox and outbox tables may be unavailable. Or an event from the source application may go undetected in ION if a business process model meant for catching the event is deactivated to perform critical changes. On such occasions you can temporarily pause a connection point. No messages are processed for that connection point and you can perform your maintenance without side effects. You can resume the message processing after your maintenance is finished and the messages for that connection point flow again.

This feature is applicable only for Infor Application Connector and Infor LN / Baan connector.

This feature has these characteristics:

- You can pause both sending and receiving of messages per connection point. These actions are independent of each other. A connection point may be paused to send new messages from its outbox while it may continue to receive messages in the inbox, or vice versa.
- A connection point remains active even if it is paused for sending or receiving. When the connection point is paused, its global status indicator, status of connectivity to the server, and its message statistics counter in the **Active Connection Points** page remain available .
- If sending or receiving messages of a connection point is paused, the connection point will only restart processing messages if you explicitly resume it. Even if a connection point is paused and subsequently deactivated, it remains paused when it is reactivated.
- Along with the number of connection points in Active or Error state, the Home Page indicates the number of connection points in paused state. This page only indicates whether a connection point is either paused for sending or receiving. From this page, you can drill down to the **Active Connection Points** page where you can view more details.

## Pausing sending messages

To pause sending messages to ION:

- 1 Select **Connect > Active Connection Points**.
- 2 Select the connection point that must be paused for sending messages.
- 3 In the **Sending Paused** column, click **Pause**. The page is automatically refreshed and the **Sending Paused** check box is selected.  
A **Resume** button becomes available.

Sending of messages by this connection point is stopped. No new messages are picked up from the source application's outbox. The connection point still processes messages that were already picked up from the outbox. This action can take some time in the background even though the sending status of the connection point is already changed to Paused.

Any new message, that was published from an application after the sending of its connection point is paused, has not yet entered ION and is not yet available in ION OneView.

## Pausing receiving messages

To pause receiving messages from ION:

- 1 Select **Connect > Active Connection Points**.
- 2 Select the connection point that must be paused for receiving messages.
- 3 In the **Receiving Paused** column, click **Pause**. The page is automatically refreshed and the **Receiving Paused** check box is selected.  
A **Resume** button becomes available.

Receiving messages by the connection point is paused. No new messages are delivered to the destination application's inbox. All new incoming messages to this application are parked in its incoming pending messages queue inside ION. The connection point still processes messages that were already

picked up from the incoming pending messages queue. This action may take some time in the background even though the receiving status of the connection point is already changed to Paused.

In OneView the time line of messages that are paused is incomplete. It indicates that a message originated from a source connection point. The receiving connection point is not part of this time line because it has not yet consumed this message.

Receiving of messages must be paused with caution. Once receiving is paused, messages start queuing up at the incoming pending messages queue of the connection point. This may result in a gradual increase of queue size. If you enable the error reporting setting, ION automatically sends a warning e-mail when this queue starts building up. For more information, see "Receiving e-mails for exceptions".

## Resuming sending / receiving messages

To resume sending or receiving messages from ION:

- 1 Select **Connect > Active Connection Points**.
- 2 Select the connection point that must be resumed for message processing.
- 3 Select one of these actions:
  - To resume the pickup of messages from an application outbox, click **Resume** in the **Sending Paused** column.
  - To resume the delivery of messages to an application inbox, click **Resume** in the **Receiving Paused** column.

The page is automatically refreshed and, depending on the selected action, the **Sending Paused** check box or the **Receiving Paused** check box is cleared.

A **Pause** button is now available.

## Resetting values for incremental load

This can be handy when using AnySQL connection point, ION API connection point or retrieve from Data Lake flow activity with incremental load.

It can happen that the output data got lost or corrupted and you must load data again. In this case you can reset incremental key values.

### AnySQL connection point

To reset incremental keys for AnySQL connection point:

- 1 Select **Connect > Active Connection Points**.
- 2 Select the connection point and click **Reset incremental**.
- 3 Reset incremental keys individually per AnySQL Model.
- 4 Click **CLOSE**

## ION API connection point

To reset last executed date time values for ION API connection point

- 1 Select **Connect > Active Connection Points**.
- 2 Select the ION API connection point and click **Reset last executed**.
- 3 Reset incremental keys individually per ION API Call.
- 4 Click **CLOSE**

## Retrieve from Data Lake activity

To rewind incremental key values for Data Lake with Retrieve from Data Lake activity:

- 1 Select **Connect > Active Document Flows**.
- 2 Select the Data Lake flow and click **Rewind incremental**.
- 3 Select relevant Data Lake activities.
- 4 Choose one of these options:
  - Reset incremental keys to original starting date and time.
  - Set incremental keys to a specific date and time.
- 5 Click **REWIND**

## Viewing or deleting pending messages

If incoming pending messages exist for an active connection point, you can view the list of messages. If messages are not flowing to a connection point, usually the first message causes this problem. You can also delete the first message in the queue.

From the list of active connection points click the drill-down button in the Incoming Pending Messages column. This shows the existing messages. You can for example see the document names of the messages.

Note that the list of pending messages displays the status at a moment in time. When the connection to the connection point is available, messages can flow through the ION Service rapidly, the messages you see in the list can be processed in the meantime.

In case a message is incorrect and blocks the processing for next messages, you can delete the first message. Select the message and click **Delete...** When deleting a message, an Error Message is created, the message is still available and can be resubmitted later.

## Resending messages

When using an application connection point of type Infor Application, messages will receive status '9' (no route) if they are sent from the connection point and not relevant for any other party.

This situation can be caused by an error in the model. For example, a document flow is incomplete, or a document flow that must be active was not activated. In that case you can still process the messages as long as they are in the application's Outbox.

To resend messages:

- 1 First activate a document flow to use the messages.
- 2 Select **Connect > Active Connection Points**
- 3 Select the connection point that has one or more 'no route' messages in the Outbox.
- 4 Click **Resend...**

The status of the messages is reset from '9' (no route) to '0' unprocessed. The messages will be sent the next time the ION Service reads the Outbox. The messages will then be delivered according to the document flow.

## Active data flows

Select **Connect > Active Data Flows**.

This page displays the active data flows.

For each active data flow the name, description and date and time of last activation are displayed.

To display more details for a data flow, select the data flow and click **Details**. Or double click the data flow.

The details page shows the model in read-only mode. You can view the properties by selecting an element in the flow. If the selected element is a connection point activity, click **Details** in the properties pane. The details of the connection point in the ION Service are shown, including the message statistics and the status.

If the selected element is a mapping activity, workflow activity, content-based routing or filter, you can view the details including the number of incoming pending messages and processed messages.

- **Incoming Pending Messages:** the number of messages that are sent to the mapping, content-based routing or filter, but not yet delivered. They are waiting because the component is still busy processing older messages.
- **Processed Messages:** the number of messages that are processed by the mapping, content-based routing or filter. Messages that are not sent to a next activity in the flow because they do not meet a filter condition are also counted as processed messages. The messages are counted since the (first) activation of the data flow.

In case of a workflow activity, you can also view:

- On the **Workflow** tab, the used workflow. Click **Details** to view the active workflow definition.
- On the **Status** tab, statistics on the number of workflows that are running, completed, canceled or failed. Click **Details** to view the corresponding workflow instances.

**Note:** The statistics contain the counters since the first activation of the data flow. Clicking **Details** takes you to the manage page of workflow instances. You will not see the workflow instances that are already archived (if multiple versions of the workflow were used, the previous versions are archived).



To view the archived workflows, navigate to the **Monitors & Workflows > Archive > Archived Workflows** page and drill down to the archived workflow instances.

## Error Messages

The **Error BODs** page in ION Desk enables you to view and handle errors. ION receives error messages sent from other applications and from ION itself. Users can therefore track error and warning conditions at the enterprise level so that all error issues can be managed together. Error messages can be received for example if:

- An application cannot process a message received from ION.
- A business rule within an application is not met during the processing of a message.
- ION received a message from an application that is not in a proper format.

The Error Message is also a regular Business Object Document identified by the verb - *Confirm* and by the noun *BOD*. "Confirm BOD" is a technical name synonymous with the term "Error Message".

## Searching error messages

You can search the Error Messages that are based on these properties:

- Its own properties.
- The properties of the original message that caused the Error Message.
- A combination of both.

To search for a specific error message or group of messages:

- 1 Select **Connect > Error BODS**.
- 2 Specify any combination of the search criteria under the **Error Message** heading:

**Message ID**

The unique identifier of the Error Message in ION.

**BOD ID**

The unique identifier of the Error Message from its Application Area.

**Sent From**

The logical ID of the sending application for the message.

**Type**

Error Message type

**Reason**

The reason description from the Error Message.

**Tenant**

Value of the Tenant that is specified in the Error Message.

**Date Range (From and To)**

Messages with a time stamp in between this range are searched. The default range is 12:00:00 AM to the 23:59:59 of the current day. Use the calendar icon to select the date. Use the time drop-down list to select the time. You can also edit the fields directly.

**Note:** The search criteria are based on date time input provided in the UTC. Ensure to convert the search criteria from your local time zone to UTC when providing the input.

- 3 Specify any combination of the search criteria under the **Original Message** heading

**Message ID**

The unique identifier of the original message in ION.

**Document ID**

The unique identifier of the original document from its Application Area.

**From**

The logical ID of the sender of the original message.

**To**

A comma separated list of logical IDs of all the receivers of the Original Message.

**Document Name**

The document Name of the original message that caused the Error Message for example: `Sync . SalesOrder`

- 4 Click **Search**.

If there are matching search results, results page(s) are displayed with a list of Error Messages. Click the column name to sort the results column in ascending or descending order. Reverse the sort order by clicking the column name again.

- 5 If there are many records that match the search, results are displayed across multiple pages. You can navigate back and forth using the arrows at the bottom of the page. Or specify the page number where you want to navigate to. You can adjust the number of records displayed per page. Use the available page size options (Fit, 25, 50, 100, 200, 500 and 1000) records per page.

- 6 You can sort each column of the list by clicking the column header - this is not applicable for the **Reason** column.

- 7 Select a particular Error Message and click **Details** to see the full details

A detail page of this Error Message is displayed. For specific information about the various fields, see [Error Message fields](#) on page 508. The **Error Messages** tab lists the error messages that are reported in the Error Message. To view the detailed reason for an error message, select the error message and click **View Reason**. The **Original Message** tab displays the information for the original message that caused the Error Message. You can format the message content or restore the view back to the original formatting. If the message content cannot be displayed or is too big, you can download it. On that tab you can also resubmit the message.

## Mark error message as handled / unhandled

After a list of the selected Error Messages is displayed, you can mark them as 'Handled' or 'Unhandled'.

- 1 Select **Connect > Error BODs**.
- 2 Specify the search criteria and click **Search**.
- 3 Select one or more Error Messages.  
Alternatively, you can select the **Select all** check box to select all Error Messages listed on the current page.
- 4 Click **Mark as Handled** or **Mark as Unhandled**.  
You can also use the search criteria to find the Error Messages to be marked as handled or unhandled. Then click **Mark All as Handled** or **Mark All as Unhandled**.  
All messages from all pages are marked as handled or unhandled.  
The actions, Mark All as Handled, Mark All as Unhandled, Resubmit All, Purge Now and Simulate Purge, can only be run one by one.

## Resubmitting the original message

Resubmitting a message sends a copy of the original message to the application that has sent the Error Message. This is particularly useful when troubleshooting errors. You can solve the situation which caused an Error Message. You can resubmit the original message to verify if the problems were resolved. Note that only one message can be resubmitted at a time.

You can only resubmit a message if the Error Message was raised by an application. If ION raises the Error Message because of an improper format, you cannot resubmit the message. In such a case, **Resubmit** in ION Desk is disabled.

**Resubmit** is enabled if the logical ID of the Error Message sender is present in the "To" list of the original message.

To resubmit a message:

- 1 Search the Error Message which original message you want to resubmit. Navigate to its detail view.
- 2 Click **View Original Message**.  
A page displaying the details of the original message opens.
- 3 Click **Resubmit**, when the situation which caused the Error Message is solved.
- 4 Specify the notes for your own administration.
- 5 Click **OK**.  
At this moment this BOD is resent to ION
- 6 To resubmit multiple messages or to resubmit without viewing the details first:
  - a Select **Connect > Error BODs**
  - b Specify the search criteria and click **Search**.
  - c Select one or more Error Messages.  
Alternatively, you can select the **Select all** check box to select all Error Messages listed on the current page.
  - d Click **Resubmit**.

You can also use the search criteria to find the Error Messages to be resubmitted. Then click **Resubmit All**. All messages from all pages are resubmitted by this action.

The actions, Mark All as Handled, Mark All as Unhandled, Resubmit All, Purge Now and Simulate Purge, can only be run one by one.

- e Note that when you resubmit multiple messages, the option to specify notes for your own administration is not available.

This activity is logged in the **Resubmit History** tab of the Error Message.

## Active Routings

Select **Connect > Active Routings**.

This page is an interactive search filter to analyze the routing configuration of ION. You can search which routings are applicable in the ION Service.

The search criteria can be based on:

- Publisher
- Subscriber
- Verb
- Noun.

If more than one of these search criteria is specified, results satisfying ALL the criteria are shown.

Searching is case-insensitive. Additionally, the search result will include return items that contain the entered value as a substring.

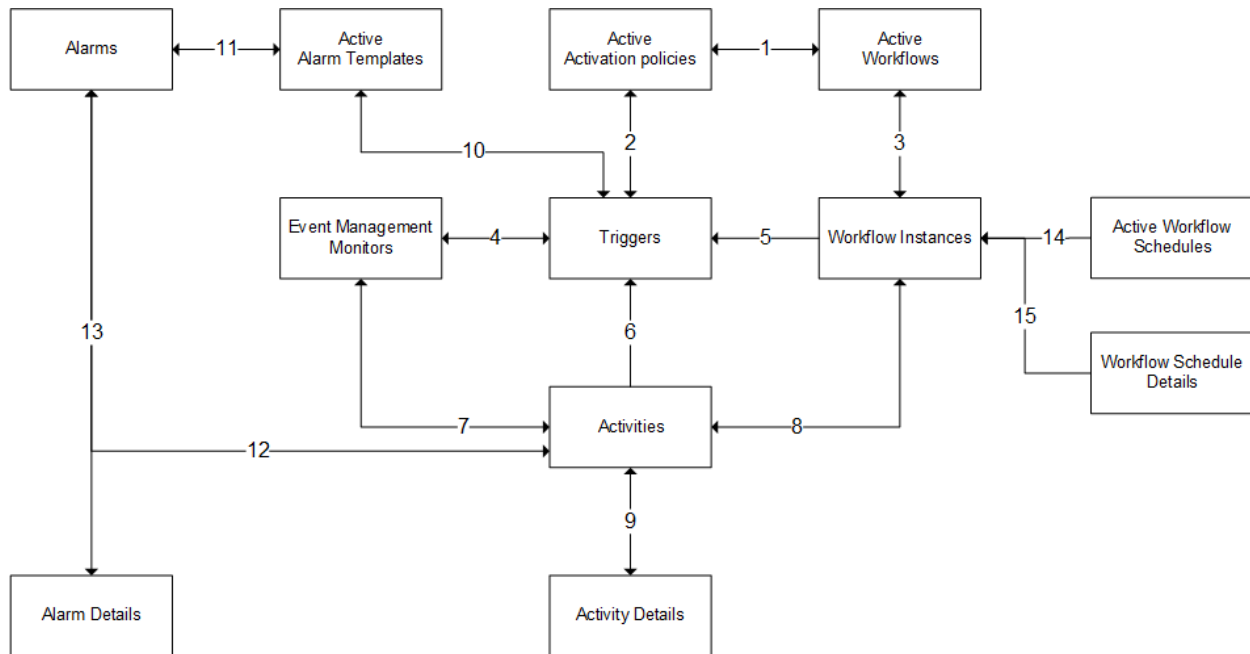
For example, when searching for verb 'sy' and noun 'cont' you can find documents Sync.ContactMaster and Sync.Contract.

Note that a publisher or subscriber does not need to be a connection point. It can also be a mapping, content-based routing or filter.

## Manage ION Process

### Navigation

This diagram shows the relations between manage pages and the corresponding navigation possibilities:



The navigation is applicable to both the management pages and the archive pages. To access these pages, select one of these menu paths:

- **Monitors & Workflows**
- **Monitors & Workflows > Activities**
- **Monitors & Workflows > Archive**

Alternatively, you can access manage pages from these components:

- The Home page. To access this page, open the menu and select **Home**.
- ION OneView.  
See [ION OneView](#) on page 361.
- The Active Data Flows page. From this page you can navigate to the manage page of the active workflows, if a document flow starts a workflow.  
See [Active data flows](#) on page 392.

The authorizations for the archive pages are the same as for the manage pages. For details on ION Desk authorizations, see the related section in this document.

This table describes the navigation possibilities in the diagram:

Number in diagram	Relation	Navigation possibilities
1	Relation between Active Activation Policies and Active Workflows pages	<ol style="list-style-type: none"> <li>1 On a selected activation policy, click <b>Workflow</b> to navigate to the Active Workflows page showing the workflow started by the activation policy.</li> <li>2 On a selected workflow, click <b>Activation Policies</b> to navigate to the Active Activation Policies page showing the activation policies for this workflow.</li> </ol>
2	Relation between Activation Policies and Triggers pages	<ol style="list-style-type: none"> <li>1 On a selected activation policy, click <b>Triggers</b> to navigate to the triggers page. This page displays all triggers for this activation policy.</li> <li>2 On a selected trigger, click <b>Activation Policy</b> to navigate to the Activation Policy page.</li> </ol>
3	Relation between Active Workflows and Workflow Instances pages	<ol style="list-style-type: none"> <li>1 On a selected workflow, click <b>Workflow Instances</b> to see the list of the existing workflow instances for this workflow.</li> <li>2 On the Workflow Instances page, click <b>Workflow</b> to navigate to the Active Workflow page showing the workflow definition for this workflow instance.</li> </ol>
4	Relation between Monitors and Triggers pages	<ol style="list-style-type: none"> <li>1 On a selected monitor in the Monitor page, click <b>Triggers</b> to navigate to the triggers page. This page displays all triggers for this monitor.</li> <li>2 On a selected Trigger, click <b>Monitor</b> to navigate to the Active Monitors page showing the monitor for this trigger.</li> </ol>
5	Relation Workflow Instances and Triggers pages	On a selected workflow instance, click <b>Triggers</b> to navigate to the Triggers page. This page shows all triggers that started this workflow instance through the activation policy.
6	Relation between Activities and Trigger pages	On a selected alert, click <b>Triggers</b> to navigate to the triggers page. This page shows all triggers that created this alert through the monitor.
7	Relation between Monitor and Activities pages	<ol style="list-style-type: none"> <li>1 On a selected monitor, click <b>Alerts</b> to view the alerts that are generated by this monitor.</li> <li>2 On a selected alert, click <b>Source</b> to navigate to the monitor that created the alert.</li> </ol>
8	Relation between Workflow Instances and Activities pages	<ol style="list-style-type: none"> <li>1 On a selected workflow instance, click <b>Tasks/Notifications</b> to see the Tasks or Notifications generated by this workflow instance.</li> <li>2 On a selected Task or Notification, click <b>Source</b> to navigate to the workflow instance that started this Task or Notification.</li> </ol>

Number in diagram	Relation	Navigation possibilities
9	Relation between Activities and Alert/Task/Notification details.	<ol style="list-style-type: none"> <li>1 On a selected item in the Activities page, click <b>Details</b> to view the details of this item.</li> <li>2 On the details page, click <b>Back</b> to return to the Activities page.</li> </ol>
10	Relation between Active Alarm Templates and Triggers pages	<ol style="list-style-type: none"> <li>1 On a selected alarm template, click <b>Triggers</b> to navigate to the triggers page. This page shows all triggers for this alarm template.</li> <li>2 On a selected trigger, click <b>Alarm Template</b> to navigate to the Alarm Templates page.</li> </ol>
11	Relation between Active Alarm Templates and Alarms pages	<ol style="list-style-type: none"> <li>1 On a selected alarm template, click <b>Alarms</b> to navigate to the Alarms page. This page shows all alarms that have been created for this template filtered by the creation date for the present day.</li> <li>2 On a selected alarm, click <b>Alarm Template</b> to navigate to the Alarm Templates page.</li> </ol>
12	Relation between Alarms and Activities pages	<ol style="list-style-type: none"> <li>1 On a selected alarm, click <b>Alert</b> to navigate to the Activities page. The button is enabled only if the alarm's status is Completed.</li> <li>2 On a selected alert, click <b>Source</b> to navigate to the alarm that generated this alert.</li> </ol>
13	Relation between Alarms and Alarm Details pages	<ol style="list-style-type: none"> <li>1 On a selected alarm, click <b>Details</b> to see its configuration information.</li> <li>2 On the details page, click <b>Back</b> to return to the Alarms page.</li> </ol>
14	Relation between Active Workflow Schedules and Workflow Instances pages	On a selected workflow schedule, click <b>Workflow Instances</b> to see the workflows started by this schedule.
15	Relations between Workflow Schedule Details and Workflow Instances pages	Click the workflow ID from the 'Started workflow' property of a schedule event to see the workflow instance with that ID.

## Navigating back

When you navigate from one Manage page to another, the page you navigate to has a Back button to return to the previous page. Alternatively, you can use the Back button of the browser or the menu by using the hamburger icon.

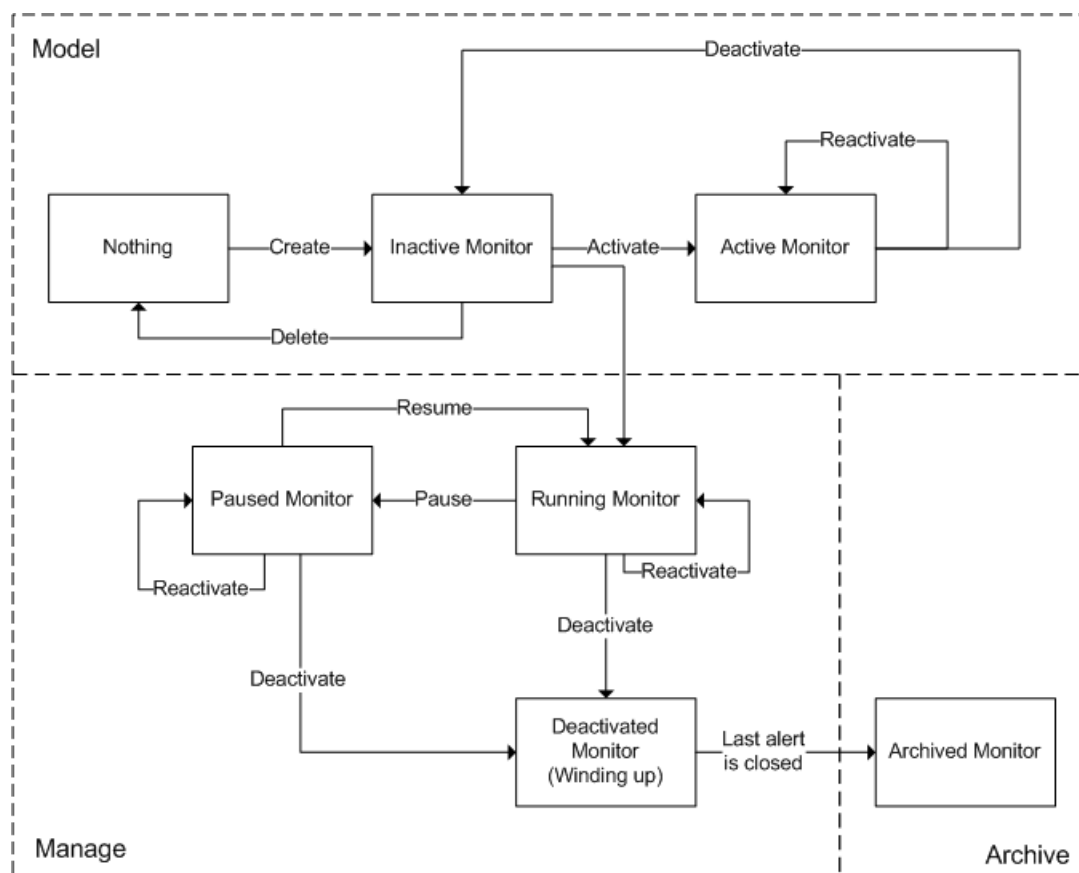
## Displaying large amounts of data

The Activities, Workflow Instances, and Trigger pages can show large amounts of data.

On these pages, pagination is used. You can change the number of records on a page or change the sorting order in the screen. To find the records that you are interested in, it may be more useful to apply a filter to narrow the amount of data, where applicable.

## State transitions

This diagram shows the life cycle of an event monitor:



The state transitions for a workflow definition or activation policy are comparable.

**Note:** If a monitor or activation policy is paused, or in error status, it can be deactivated.



# Manage Event Management

## Viewing the status of Event Management

To view a summary of the status of Event Management, select **Home**. On the Event Monitors tile, the number of active and paused event monitors is displayed. Click the headline of the Event Monitors tile to view a summary grid with the active monitors. Optionally, click a counter to drill down to the corresponding manage page.

You can also navigate to the summary grid from the menu by selecting **Monitors & Workflows > Active Monitors**.

## Viewing active monitors

To view status information on activated monitors:

- 1 Open the **Active Monitors** management page.

This information is displayed:

**Monitor Name**

The unique name that identifies the monitor.

**Version**

The version of the monitor model. The version number is increased each time the monitor is deactivated, changed, and activated again.

**Activation Date**

The date and time the monitor was activated.

**Status**

The status of the monitor. Possible values:

- **Running**  
The monitor is busy. You can pause a running monitor.
- **Paused**  
The monitor is temporarily stopped. You can resume a paused monitor.
- **Error**  
The monitor is stopped because of an error.
- **Winding up**  
The monitor is deactivated, but still has alerts open. The monitor remains visible in the manage pages until all alerts are closed or canceled. After that, the monitor is automatically moved to the archive pages.

**Documents Processed**

The number of documents the monitor has evaluated since its activation.

### Alerts Created

The number of alerts this monitor has generated since its activation.

You can filter the list of active monitors by **Monitor Name**, **Status**, **Activation Date**, and **Deactivation Date**.

- 2 To view the details of an active monitor, select the monitor and click **View Model**.  
If the monitor was changed and activated again, several variants are displayed and you must select one of them to see. A read-only view of the monitor model is displayed. Click the **Back** icon button to return to the list of active monitors.

## Pausing active monitors

In the **Active Monitors** management page, select the monitors to suspend and click **Pause**.

The monitors stop evaluating incoming documents and sending alerts.

**Note:** All data recorded by a monitor, such as the multiple occurrence counter and elapsed waiting time, is kept. The saved data is processed when you resume the monitor. See the example.

### Example

A monitor is created that sends an alert when a `SalesOrder` with status canceled is published. If you Pause this monitor, sales order documents are still evaluated, but no alerts are sent. When the monitor is Resumed, the alerts are sent and the monitor continues execution.

## Resuming monitors

In the **Active Monitors** management page, select the monitors to resume and click **Resume**.

The monitors restart evaluating incoming documents and sending alerts.

**Note:** When you resume a monitor, the monitor continues where it was stopped. That is: the monitor continues with the data that was kept when the monitor was suspended.

## Showing alerts

To show the alerts that are created by a monitor:

In the **Active Monitors** management page, select the monitor for which to view alerts and click **Show Alerts**.

The **Activities** page with all alerts for the selected monitor is displayed.

## Showing triggers

To show the triggers (BOD data) that are processed by a monitor:

- 1 Go to the **Active Monitors** management page.
- 2 Select the monitor for which to view triggers.
- 3 Click **Triggers**.

The page with all triggers for the selected monitor is displayed.

## Viewing triggers

You cannot directly navigate to the **Triggers** page. To display the **Triggers** page, click **Triggers** on these pages:

- Manage monitor page
- Manage alarm templates page
- Manage activation policies page
- Manage workflow instances page
- Manage activities page

See [Navigation](#) on page 396.

The **Triggers** page is a master-details page. The list of triggers is displayed in the upper part of the screen. If a trigger is selected, its details are displayed in the lower part of the screen.

When the logging of triggers is disabled, a notification is displayed above the triggers list.

You can change the settings for logging triggers through this ION Desk menu: **Configuration > ION Service > Properties > Event Management engine**

The list of triggers shows this information:

### **Creation Date**

The date when the trigger was created.

### **Monitor**

The name of the monitor that processed this trigger.

### **From LogicalID**

The application that sent the trigger.

### **Type**

The noun name of the trigger.

### **Document ID**

The unique document identifier of the trigger.

### **Revision ID**

The revision identifier from the document identifier, if it was present in the incoming message.

### **Accounting Entity**

The accounting entity of the trigger.

**Location**

The location of the trigger.

**Ignored**

Indicates whether the trigger was processed by the monitor.

**Alert ID(s)**

The identifiers of the alerts that the monitor has created based on this trigger.

The trigger details show a tree structure of the trigger attributes with the attribute values and its datatype.

## Viewing archived monitors

Monitors that are deactivated, and do no longer have any open alerts, are archived.

To view the archived monitors:

- 1 Select **Monitors & Workflows > Archive > Archived Monitors**.

- 2 View the monitor details and the archived alerts.

The pages are comparable to the manage pages described above. The pages are view-only; you cannot perform any action on archived data.

## Manage Alarm Templates

### Viewing the status of alarm templates

To view a summary of the status of alarms, select **Home**. On the **Alarm Templates** tile, the number of active and paused alarm templates is displayed. Click the headline of the **Alarm Templates** tile to view a summary grid with the active templates. Optionally, click a counter to drill down to the corresponding manage page.

You can also navigate to the summary grid from the menu by selecting **Monitors & Workflows > Active Alarm Templates**.

### Viewing active alarm templates

To view status information on activated templates:

- 1 Open the **Active Alarm Templates** management page.

This information is displayed:

**Alarm Template Name**

The unique name that identifies the alarm template.

**Version**

The version of the alarm template model. The version number is increased each time the template is deactivated, changed, and activated again.

**Activation Date**

The date and time the alarm template was activated.

**Status**

The status of the alarm template. Possible values:

- **Running**  
The alarm template is busy. You can pause a running alarm template.
- **Paused**  
The alarm template is temporarily stopped. You can resume a paused alarm template.
- **Error**  
The alarm template is stopped because of an error.

**Documents Processed**

The number of documents the alarm template has evaluated since its activation.

**Alerts Created**

The number of alerts that have been created by alarms generated for this template since its activation.

You can filter the list of active alarm templates by **Alarm Template Name**, **Status**, **Activation Date**, and **Deactivation Date**.

- 2 To view the details of an active alarm template, select the template and click **View Model**.  
If the alarm template was changed and activated again, several variants are displayed and you must select one of them. A read-only view of the alarm template model is displayed. Click the **Back** icon button to return to the list of active alarm templates.

## Pausing active alarm templates

In the **Active Alarm Templates** management page, select the alarm templates to suspend and click **Pause**. The alarm templates stop evaluating alarm instances for incoming documents.

**Note:** The data from the incoming documents is kept. The saved data is processed when you resume the alarm template. See the example.

---

An alarm template for SalesOrder is Paused, while there is an active alarms checking for SalesOrder with status Canceled. If a SalesOrder with status Canceled is received during the time when the template is paused, the information is kept until the template resumed. If at that time the alarm is still active, an alert is sent and the alarm status is set to Completed. If the alarm was stopped by the Alarms user, or its lifetime has expired while the template was paused, no alert is created.

## Resuming alarm templates

In the **Active Alarm Templates** management page, select the alarm templates to resume and click **Resume**. The alarm templates restart evaluating documents and executing alarms.

## Showing alarms

To show the alarms that are created for an alarm template:

- 1 Go to the **Active Alarm Templates** management page.
- 2 Select the alarm template for which you want to view alarms.  
The **Alarms** page with the alarms for the selected alarm template is displayed. By default, the list is filtered to show only the alarms that are created in the present day.
- 3 Change the filter as required to see the whole list.
- 4 Click **Alarms**.

## Showing triggers

To show the triggers (BOD data) that are processed by an alarm template:

- 1 Go to the **Active Alarm Templates** management page.
- 2 Select the alarm template for which to view triggers.
- 3 Click **Triggers**.  
The page with all triggers for the selected alarm template is displayed.  
For details, see [Viewing triggers](#) on page 403.

## Viewing archived alarm templates

Alarm templates that are deactivated are archived.

To view the archived alarm templates:

- 1 Select **Monitors & Workflows > Archive > Archived Alarm Templates**.
- 2 View the alarm template details and the corresponding alarms.  
This page is view-only. You cannot perform any action on the archived alarm templates. The alarms that have been created for an archived alarm template are displayed in the **Monitors & Workflows > Alarms** page, based on the template name and version.

## Manage ION Workflow

### Viewing the status of workflows

- 1 To view a summary of the status of Workflow in ION, select **Home**.  
Among other things, the page shows the number of:
  - Active workflows
  - Active and paused activation policies
  - Activities
  - Active workflow schedules
- 2 On each tile, click the headline or the counter in the tile drill-down to the corresponding manage page.
- 3 Alternatively you can navigate directly to the workflow manage page by selecting **Monitors & Workflows > Active Workflows**.

### Viewing the status of a workflow

- 1 Open the **Active Workflows** management page.
- 2 The page contains this information:

**Workflow Name**

The unique name that identifies the workflow definition.

**Version**

The version of the workflow model. The version number is increased each time a workflow is deactivated, changed, and activated again.

**Activation Date**

The date and time the workflow was activated.

**Deactivation Date**

The date and time the workflow was deactivated. This field is empty as long as the workflow is active.

**Status**

One of these statuses:

- Running  
The workflow definition is active.
- Winding up  
The workflow definition is deactivated, but it still has running workflow instances. The workflow remains visible in the manage pages until all workflow instances are finished or canceled. After that, it is automatically moved to the archive pages.

**Running**

The number of workflow instances that are currently running.

**Completed**

The number of workflow instances that successfully finished.

**Canceled**

The number of workflow instances that were canceled.

See [Canceling workflow instances](#) on page 317.

**Failed**

The number of workflow instances that stopped because of a failure.

**Total**

The total number of workflow instances that were started for the workflow definition.

**Total = Running + Canceled + Failed + Completed.**

You can filter the list of active workflows by **Workflow Name**, **Status**, **Activation Date**, and **Deactivation Date**.

- 3 Optionally, complete these steps:
  - a To view the details of an active workflow definition, select the workflow and click **View Model**. If the workflow was changed and activated again, several variants are displayed and you must select one of them to see. A read-only view of the workflow model is displayed. Click the **Back** icon to return to the list of active workflows.
  - b To view the activation policies for a selected workflow, click **Activation Policies**.
  - c To view the instances for a selected workflow, click **Workflow Instances**.  
See [Viewing the status of a workflow instance](#) on page 408.

For details about navigation between pages, see [Navigation](#) on page 396.

## Viewing the status of a workflow instance

- 1 In the **Active Workflows** management page, select a workflow in the list and click **Workflow Instances**.
- 2 The workflow instances page lists all instances for the selected workflow. The default filter is to show only the workflow instances that were created since the workflow activation date till the current day.  
If required, adjust the filter and click **Search** to refresh the list of workflow instances. The total number of instances that were found according to the search criteria is displayed at the bottom of the screen. Use the paging control to navigate through the complete list. Alternatively, you can search for a particular workflow instance by its ID.  
You can generate a CSV report for a list of workflow instances that results from a search query. See [Generating reports for workflow instances](#) on page 422.
- 3 In the workflow instances page, view this information about instances that were started for the selected workflow:



**ID**

The numeric ID that identifies the workflow instance.

**Instance Name**

The name of the workflow instance, if specified at the time the instance was started. The default value is the workflow definition name.

**Start Time**

The date and time the instance was started.

**Status**

This field displays one of these statuses:

- **Started**  
The instance is currently running. If necessary, you can cancel a running instance. See [Canceling workflow instances](#) on page 317.
- **Completed**  
The instance finished successfully.
- **Canceled**  
The instance was canceled by an application user or a system administrator.
- **Failed**  
The instance stopped because of a failure.

**End Time**

The date and time the instance stopped.







**Reason**



This field is specified only for canceled instances. The field shows the reason why the instance was canceled.

**Start Source**

The source that started this workflow instance.

This table shows the source type and the source identifier:







Icon	Source type	Identifier
	Activation Policy	Activation policy name
	Alert	Alert ID
	API	Logical ID of the application which called the API
	Process.Workflow BOD	MessageId of the BOD
	User Action	User Name
	Document Flow	Message ID of the Process.Workflow BOD sent from Document Flow

Icon	Source type	Identifier
	Workflow	Workflow Name
	Workflow Schedule	Workflow schedule name

### Cancel Source

The source that canceled this workflow instance.

This table shows the source type and the source identifier:

Icon	Source type	Identifier
	Activation Policy	Activation policy name
	API	Logical ID of the application which called the API
	Process.Workflow BOD	MessageId of the BOD
	User Action	User Name
	ION Desk Admin Action	The value "ION Desk Admin" is displayed
	Workflow Schedule	Workflow schedule name

#### 4 Optionally:

- a To view the Tasks and Notifications for a selected workflow instance, click **Tasks/Notifications**.
- b To view the workflow for a selected workflow instance, click **Workflow**.
- c To view the activation policies for a selected workflow instance, click **Activation Policies**.
- d To view the document flow that started a selected workflow instance, click **Document Flow**.
- e To view the triggers for a selected workflow instance, click **Triggers**. This option is only available for workflows started by an activation policy.

For details about navigation between pages, see [Navigation](#) on page 396.

## Viewing archived workflows

Workflows that are deactivated, and do no longer have any running workflow instances, are archived.

To view the archived workflows:

- 1 Select **Monitors & Workflows > Archive > Archived Workflows**.
- 2 View the workflow details and the archived workflow instances.

The pages are comparable to the manage pages described above. The pages are view-only; you cannot perform any action on archived data.

## Manage workflow activation policies

You can access the management pages for workflow activation policies by clicking on the **Activation Policies** tile from the **Home** page, using ION OneView. Alternatively, select **Monitors & Workflows > Active Activation Policies**.

### Using ION OneView

ION OneView provides visibility for all documents that passed through the ION Service. This includes the documents that cause a workflow to be started or canceled.

You can use OneView, for example, to perform these tasks:

- View the ProcessWorkflow documents that were sent to the workflow engine.
- View the documents that were sent to an activation policy.

For details, see [ION OneView](#) on page 361.

### Viewing the status of an activation policy

- 1 Open the **Active Activation Policies** management page.

In the upper part of the **Activation Policies** page, this information is displayed:

**Name**

The unique name that identifies the activation policy.

**Policy Type**

This field displays one of these types:

- **Start**  
The policy is used to start a workflow.
- **Cancel**  
The policy is used to cancel a workflow.

**Version**

The activation policy version. The version number is increased every time the activation policy is deactivated, changed, and activated again.

**Workflow and Workflow Version**

The workflow that is, or will be, started or canceled by the policy.

**Activation Date**

The date and time the policy was activated.

**Status**

The status of the activation policy. This field displays one of these statuses:

- **Running**  
The activation policy is currently running.

- **Paused**  
The activation policy is paused.
- **Winding up**  
The activation policy is deactivated, but still has alerts open. The activation policy remains visible in the manage pages as long as the corresponding workflow definition is also visible in the manage pages. When the workflow definition is also deactivated and all started workflow instances are completed, the workflow and the activation policy are automatically moved to the archive pages.

### **Documents Processed**

The number of documents that are processed by the activation policy.

### **Events**

The number of events that happened. An event is the starting of a new workflow instance or the cancelation of a running workflow instance.

You can filter the list of active activation policies by **Activation Policy Name**, **Status**, **Activation Date**, and **Deactivation Date**.

You can Pause or Resume the selected activation policy or policies. If the activation policy is Paused, no workflows are started or canceled by the activation policy. The incoming BODs are still monitored. Therefore, after resuming the activation policy, workflows are started or canceled for any documents that arrived in the meantime, if these documents match the activation policy rule.

- 2 To view the details of an active activation policy, select the policy and click **View Model**.  
If the activation policy was changed and activated again, several variants are displayed and you must select one of them to see. A read-only view of the activation policy model is displayed. Click the **Back** icon button to return to the list of active activation policies.

## Viewing archived activation policies

An activation policy is archived when it is inactive, the used workflow is inactive, and no running workflow instances exist anymore for the workflow.

To view the archived activation policies:

- 1 Select **Monitors & Workflows > Archive > Archived Activation Policies**.
- 2 View the archived activation policies.

The archive pages are comparable to the manage pages described above. The pages are view-only; you cannot perform any action on archived data.

## Manage workflow schedules

To access the management pages for the workflow schedules, click the **Workflow Schedules** tile on the ION Desk **Home** page.

Alternatively, in ION Desk, select **Monitors & Workflows > Active Workflow Schedules**.

On the **Active Workflow Schedules** page, a list of all schedules that are known in the runtime is displayed. A schedule can have one of these statuses:

- **Running** – the schedule is active. It will run at the next run date according to its schedule frequency.
- **Paused** – the schedule is active, but at the next run date it will not attempt to start a workflow.
- **Completed** – the schedule has reached the configured end date and time or it has reached the maximum number of runs.

For an active workflow schedule, you can perform these actions:

- Click **Pause** to move a running schedule to the Paused status.
- Click **Resume** to move a paused schedule to the Running status.
- Click **Workflow Instances** to see all workflow instances that were started by this schedule since it was activated.
- Click **Details** to see more information about this schedule and a list of all events that happened since this workflow schedule was activated.

## Manage alarms

To view alarms that were created by all users, use the **Monitors & Workflows > Alarms** page. This page shows all alarms that were created for all templates that have been active in the system. By default, the list is filtered by creation date of the alarms to show only the alarms from the present day.

On the **Monitors & Workflows > Alarms** page, you can perform these actions:

- View a filtered list of alarms.
- For one selected alarm, view its details and run these commands: **Show Alarm Template**, **Show Alert**, and **Stop**.

To filter the list of alarms:

- 1 Specify this information:

### **From Start Date/ To Start Date**

Specify a start date and an end date to search on. Only alarms that were created within this date-time range are displayed.

### **Status**

By default, all statuses are displayed. Select a specific status value to limit the search.

### **Template**

Specify a template name or a part of a name to find only alarms that are created for this template.

### **Document**

Specify a document name or part of a name to find only alarms created by templates that monitor this particular document.

### **Author**

Specify a user's full name or part of a name to filter only alarms created by this user.

- 2 Click **Search** to apply the filter options.
- 3 Alternatively, you can search for an alarm by its ID.

## Viewing search results

These properties are displayed for a list of alarms resulting from a search filter:

### **ID**

The identifier of this alarm instance. This ID is generated by the system for each alarm.

### **Template**

The name of the template from which this alarm was created.

### **Version**

The version of the template from which this alarm was created.

### **Document**

The document name that is used in the template from which this alarm was created.

### **Alarm Name**

The name of the alarm as it was defined by the user. By default, the alarm name is generated using the template name and the timestamp at the creation time. A user may give an alarm any name. The same alarm name might be used by several users or for different templates.

### **Author**

The name of the user who created this alarm.

### **Status**

The status of this alarm. Possible values:

Status	Description
Active	The alarm is running and waiting for documents to be evaluated for its condition
Expired	The alarm lifetime expired before a document was detected that should create an alert
Completed	The alarm was triggered by a document which caused one alert to be created
Stopped	The alarm was stopped by a user
Canceled	The template was deactivated while the alarm was running.

### **Start Date**

The date when an alarm was started. This is also the time from which alarms start to show up in the **Monitors & Workflows > Alarms** page.

### **Completion Date**

The date when the alarm finished execution.

**Lifetime**

The lifetime of the alarm expressed in days.

## Viewing alarm details

Select one alarm to view its details and click **Details**.

This information is displayed in the details screen:

- Alarm information in the left panel:
  - The same properties as displayed in the alarm list: **ID, Template, Version, Document, Alarm Name, Author, Status, Start Date, Completion Date, Lifetime**.  
For a description of these properties, see [Viewing search results](#) on page 414.
  - The **Alert ID** property. If the alarm has status Completed, this property contains the identifier of the alert that was generated by this alarm. This identifier is automatically generated by the system
  - The **Message** property. This is the message string as typed by the user and is used as the alert message. It may contain parameter placeholders that are enclosed by square brackets, []. It may be ended by a category marked by ##.
- Alarm information in the right panel
  - The **Attributes** tab shows attributes that are used in the alarm. The attributes that are used in conditions and the attributes that are selected for the alert content are displayed in the same list. You can view the attribute name as it was given in the template and the attribute data-type.
  - The **Conditions** tab shows the list of conditions that are used by this alarm. The conditions are joined automatically by logical AND. Therefore, the alarm creates an alert when all conditions evaluate to true.
  - The **Distribution** tab shows the list of distribution items as it was defined by the user who created the alarm. The list contains the user and group identifiers and the flag to send email.

## Stopping an alarm

You can stop alarms that have the Active status.

- 1 On the **Monitors & Workflows > Alarms** page, select the active alarm to be stopped.
- 2 Click **Stop**.

The alarm stops evaluating documents and its status changes to Stopped.

## Pulse activities

Activities can be alerts, tasks, and notifications. Alerts are created from an event monitor, or directly from an application by sending a BOD. Tasks and notifications are created from a workflow, or directly from an application by sending a BOD.

Administrators can handle activities in manage pages in ION Desk. End users can see and handle activities in Infor Ming.le page, the Infor Inbox application, or the homepage widgets for alerts, tasks and notifications.

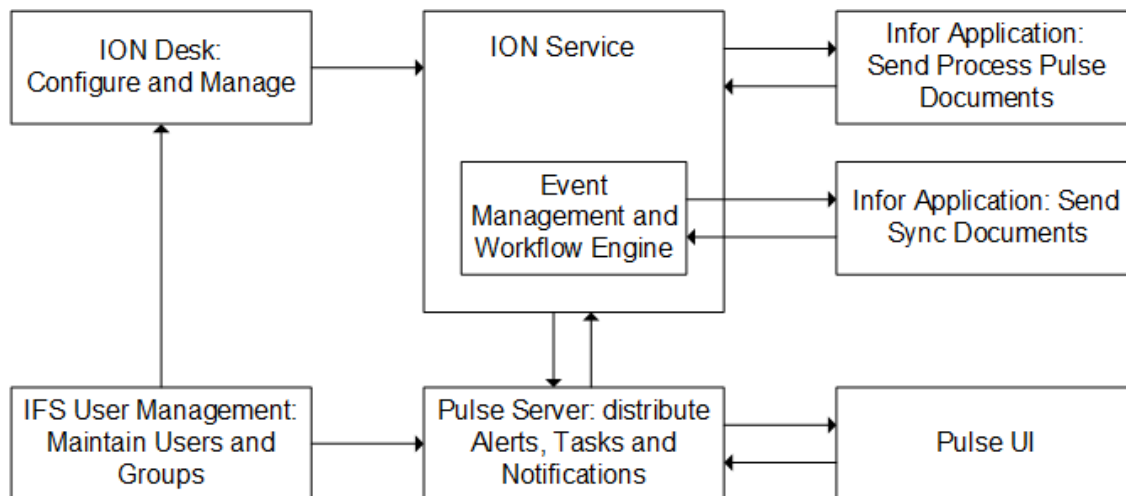
## Concepts

### Pulse and activities application architecture

The Pulse engine part on the server side runs as engine in the ION Service container. The Pulse engine is accessed by the Event Management and Workflow engines and pushes tasks and alerts through secured web-service calls. Other Infor applications can also send `PulseAlert`, `PulseTask`, and `PulseNotification` messages directly to Pulse.

For details, see the *Infor ION Development Guide*.

The Pulse UI acts as a client application for Pulse and displays the activities and workflow information relevant to each user.



When a new alert or task is created, it is visible for all users from the distribution list of this alert and task. If the distribution list contains group names, all users that are part of this group receive a notification when the task or alert is created.

Tasks and alerts are shown in these locations:



- The alerts or tasks widgets from the Infor Ming.le homepages, the activity feed page, and the Infor Inbox application in Infor Ming.le of the user to which the task or alert is assigned.
- If the alert is not assigned to a user: The alerts or tasks widgets from the Infor Ming.le homepages, the activity feed page in Infor Ming.le and the Infor Inbox application of all users who are in the distribution list of the task or alert.
- The activities pages in ION Desk.

## Management

### Managing activities

To view alerts and tasks of all users, business process administrators and system administrators can use the **Monitors & Workflows > Activities** menu option. In this page, they can view all tasks, alerts, and notifications in the system. All other users can access only their personal task list.

On the **Monitors & Workflows > Activities** page you can perform these actions:

- View all alerts/tasks/notifications in the system.
- Search for an alert, task, or notification by its ID.
- Redistribute any alert or task. You cannot redistribute a notification.
- Generate a report for the activities from the search criteria.
- Cancel one or all activities from the search criteria, with the exception of Tasks from Workflow.

System Administrators can use ION Desk to open the Manage pages of the Activities module.

To manage alerts, tasks, and notifications:

- 1 Open the **Activities** page. This page shows the **Activities** view with all the activities from the present day.
- 2 Specify the search criteria for the pulse items to be displayed:

**Type**

The type of pulse item, such as Alert.

**From/To**

The date and time that the alert, notification, or task was generated. Specify a date from the calendar. Specify a time from the clock.

**Source**

The name of the monitor or workflow that generated the alert, notification, or task.

**Status**

The status of the alert, notification, or task. These are the possible statuses:

- **All**  
List all statuses of the alert, task, or notification.
- **New**

The alert, task, or notification is created and distributed to users. No user has worked on the alert or task.

- **Assigned**

The alert or task was manually assigned to a user or to a role.

- **Unassigned**

The alert or task was manually unassigned.

- **Done**

A user completed the alert, task, or notification. The alert, task, or notification can no longer be changed.

- **Canceled**

A Task has status canceled if the workflow instance that generated this Task was canceled. An Alert and a Notification have status canceled if the system administrator manually canceled them.

### Assignee Person ID

The user to which the alert or task is assigned. Use the search dialog box to find a user by the name. After selected, the IFS-ION Person ID of this user is saved and used in the search query. For information on user accounts, see the Infor Federation Services documentation.

- 3 Click **Search**. A list of all alerts, tasks, and notifications that fulfill the search criteria is displayed. You can sort on each column of the list by clicking the column header. Sorting works on all results matching the specified search criteria. You cannot sort on Type and Source Type. Paging is now introduced, no limit of number of items that can be viewed. At the bottom of the page, the total number of found items for these search criteria is displayed.

If an alert or task is escalated, a red icon is displayed next to the status label in the **Status** column. You cannot filter only the activities that are escalated.

- 4 Select the event of interest.

Several actions are available for the selected event:

- Click **Details** to view the event details.
- Monitor and Triggers can be shown for an alert created by Event Management.
- The Workflow Instance can be shown for a task or notification created by Workflow.
- You can click **Cancel** for alerts, notifications and tasks that are created by `Process.Pulse Task BOD`.

- 5 Click **Details** to drill down to view the details of the selected event.

Depending on the activity type and depending on its source, generic information is shown.

This table shows the generic information:

Property	Applies to	Description
Type	Alert, Task, Notification	The activity type: Alert, Task or Notification.
Source Type	Alert, Task, Notification	The source type that generated this activity, such as a workflow for a task or a monitor for an alert.

Property	Applies to	Description
Source	Alert, Task, Notification	The identifier or the name of the source of this activity. For example, the name of a monitor that generated an alert.
Context ID	Task, Notification	If the task or notification was generated by a workflow, this is the workflow instance ID.
Sub Context ID	Task, Notification	If the task or notification was generated by a workflow, this is the ID of the workflow step.
Status	Alert, Task, Notification	The activity status such as New, Assigned, Done, etc.
Priority	Task	The task priority as it was configured when the task was created. The priority can be: High, Medium, Low.
Escalated	Alert, Task	If the task or alert is escalated, this flag is ON.
Assignee	Alert, Task	Contains the name of the user to whom this activity was assigned to.
Creation Date	Alert, Task, Notification	The date and time when this activity was created, displayed in local time.
Due Date	Alert, Task	If a due date was configured for this task or alert, it is displayed here as local time.
Message	Alert, Task, Notification	The message default text as it was configured for the activity.

**6** On the **Details** tab, view the parameters and structures of the alert, task, or notification.

The parameters can differ per alert, task, or notification. These parameters depend on the definition of the monitor or workflow that generated the alert, task, or notification. Structures are available for tasks and notifications. They can be configured in the workflow model. Both parameters and structures are shown with their name from the workflow definition, and not with the labels. The structure in a task or notification in Infor Ming.le and how the structure name is displayed here can be interpreted differently by the user..

If custom drillbacks are generated for an alert, these are also visible in the **Details** tab. The values that are passed to the view parameters are shown.

If a custom form view is used to show the details of a task, the drillback that is generated for the form is displayed.

**7** On the **Notes** tab, view the notes that are entered for the alert or task. Here you can also view propagated notes if they are included in a task or a notification created by Workflow.

The tab shows this information:

**Date**

The date and time that the note was specified.

**User**

The user who specified the note.

**Note**

The text that was specified by the user. This text can be up to 4000 characters. If the text is not shown completely in the grid, click **Zoom** for each note to read the full popup text.

**Is Propagated**

If the tasks are generated by Workflow, the notes displayed are the ones added to this particular step. These notes are propagated from previous steps in this workflow. If the notifications are generated by Workflow with the option to include previous notes. The notes displayed are the ones that are propagated from the previous steps.

- 8 Click the **Distribution List** tab and view the current distribution list.

To re-distribute the alert or task to another distribution list:

- a Click the expand button next to the **Redistribute** label. A new set of buttons is displayed.
- b Edit the distribution list using **Add** and **Remove**.
- c When the new distribution list is completed, click **Apply**.

The alert or task is sent again to the users in the new distribution list. The alert or task keeps its assigned or unassigned status after it was redistributed, except when the current assignee is removed from the distribution list - in this case, it becomes unassigned. To discard changes and return to the initial distribution list, click **Discard**. This does not resend the alert or task to the users.

- 9 Click the **Distribution Users** tab and view the list of users from the distribution list. The distribution list can contain a group name. The tab shows all users that belonged to that group at the moment the alert, task, or notification was sent.

For details on users and groups, see the *Infor Federation Services Administration Guide (ifsag)*.

## Searching for an activity by ID

- 1 Turn the **Search by ID** switch on.
- 2 Specify the ID number to find.
- 3 Click **Search**.

If the ID exists, the activity with this ID is displayed in the search results.

## Canceling outstanding activities

If too many tasks, alerts, and notifications are present in the system, you can cancel them from the **Monitors & Workflows > Activities** page. The canceled activities are not shown to users of Infor Ming.le anymore. Canceled alerts, tasks, and notifications are not removed from the Pulse engine database and they can still be seen in the Manage or Archive pages.

To cancel tasks generated by Workflow, you must cancel the running workflow instances that generated those tasks.

See [Canceling a workflow instance in ION Desk](#) on page 317.

## Canceling activities in a batch

- 1 Open the menu and navigate to the **Activities** page.
- 2 Edit the filter to find the items to be canceled and click **Search** .  
The **Cancel All** button is active and has effect on all activities in the search results.
- 3 Click **Cancel All** to run a batch cancel.  
A list of the total number of activities that are found from the current search is displayed.
- 4 Click **Yes** to continue  
After the batch cancellation is finished, the total number of cancelled activities is displayed. Tasks generated by Workflow and Alerts, Tasks and Notifications that are already Done or Canceled, cannot be canceled again. These cancellations are not included in the final count.  
The status of the activities becomes *Cancelled* where applicable.

## Canceling activities individually

- 1 In ION Desk, select **Monitors & Workflows > Activities**.
- 2 Edit the filter to find the activities to be canceled and click **Search**
- 3 Select the alerts and notifications to be cancelled. The **Cancel** button becomes active.
- 4 To cancel the selected activities, click **Cancel** and reply **Yes** in the confirmation dialog. The status of the activities becomes *Cancelled*.

## Viewing archived activities

If a monitor or workflow definition is archived, the corresponding alerts, tasks, and notifications are also archived.

To view these archived items:

- 1 Select **Monitors & Workflows > Archive > Archived Activities**.
- 2 View the archived alerts, tasks, and notifications.

Alternatively, you can search for an archived activity by its ID.

## Generating reports for user actions

For each activity, the status trail of user actions is recorded. To review all user actions within a time interval, you can generate reports in the form of PDF documents.

You can generate reports from these pages:

- **Monitors & Workflows > Activities**
- **Monitors & Workflows > Archive > Archived Activities**

- 1 Specify a filter.

See [Managing activities](#) on page 417 and [Viewing archived activities](#) on page 421.

- 2 Click **Search** to apply the filter.
- 3 If there are activities that fulfill the filter conditions, you can click **Generate Report** to generate a report for the activities included in the current filter.
- 4 To confirm the report generation, in the **Create Report** dialog box, click **Yes**.
- 5 If an error message about "font not available" is displayed, contact your Infor representative.  
**Note:** Alerts, Tasks, and Notifications that do not have status "Done" or "Cancelled" are closed after the period of time included in the report. To verify the complete item history, check each item individually, or generate the report again with the same "From date" and with "To date" later in time.

## Generating reports for workflow instances

Multiple workflow instances can be started for each workflow, each having a different set of input parameters. To get an overview of a set of workflow instances for a given workflow definition version, you can generate a CSV report. For active workflows, a status report or an aging report can be generated. For archived workflows a status report can be generated.

You can generate the workflow instances report from these pages:

- **Monitors & Workflows > Active Workflows > Workflow Instances** - for one selected workflow
  - **Monitors & Workflows > Archive > Archived Workflows > Workflow Instances** - for one selected workflow
- 1 Specify a filter.  
 See [Viewing the status of a workflow instance](#) on page 408 and [Viewing archived workflows](#) on page 410.
  - 2 Click **Search** to apply the filter.
  - 3 Click **Generate Report** to generate a CSV report for the workflow instances in the current filter.  
 For active workflows, select the **Create the workflow status report based on the current filter** option to generate the status report.  
 Select the **Create the workflow aging report for the running workflow instances from the current filter** option to generate the workflow aging report.  
**Note:** Only a maximum of 1000 workflow instances can be included in a report. If required, adjust the filter to create several reports for more workflow instances.
  - 4 To confirm the report generation, in the **Create Report** dialog box, click **Yes**.
  - 5 When the report is ready, it is displayed on the **Reports** page. The name of the status report is `WorkflowInstancesReport.csv`.  
 The name of the aging report is `WorkflowAgingReport.csv`.  
 See [Downloading reports](#) on page 426.  
 For details about the information that is included in a report about workflow instances, see these sections:
    - [Columns in the workflow status report](#) on page 423
    - [Workflow instance status information](#) on page 426

## Columns in the workflow status report

This table shows the columns in the workflow status CSV report:

Column Name	Description
InstanceID	Workflow instance ID. Data type: string.
Name	Workflow definition name. Data type: string.
Version	Workflow definition version. Data type: number.
Status	Workflow instance status. Possible values: Running, Completed, Canceled, Failed.
StartTimeUTC	Workflow instance start time. Date and time are displayed as UTC.
EndTimeUTC	Workflow instance end time, if available. Date and time are displayed as UTC.
StartSourceType	Workflow start source type. Possible values: BOD, Manual, Alert, Workflow, Activation Policy, Scheduler.
StartSourceName	Workflow start source name. Depending on the source type, this is an identifying property of the source.
CancelSourceType	Workflow cancel source type. Similar to the start source types.
CancelSourceName	Workflow cancel source name. Similar to the start source name.
StepID	Workflow step ID. Data type: number.
StepType	Workflow modeler step type.

Column Name	Description
PulseID	<p>In case of a step of type User Task, this is the Pulse ID of the task.</p> <p>In case of a workflow step of type Parallel Task, multiple copies are created as individual tasks. In this case, the <b>PulseID</b> column contains the list of Pulse IDs of all individual tasks that are still open. The list is delimited by the pipe " " symbol. The IDs of the individual tasks that are already closed are not included in the report.</p> <p>In case of a workflow step of type Task Chain, the <b>PulseID</b> column contains the ID of the task that is currently open.</p> <p>You can use this ID to find the task on the <b>Manage Activities</b> page in ION Desk.</p>
StepName	Workflow step name.
StepStatus	Possible values: WAITING, CANCELLED, COMPLETED.
<PARAM:parameter name from workflow model>	The CSV report contains a separate column for each parameter in the workflow. The names of these columns contain the parameter names that are specified in the workflow model, and the prefix PARAM:. The columns are sorted based on the alphabetical order of the parameter names.

## Columns in the workflow aging report

This table shows the columns in the workflow aging CSV report:

Column Name	Description
Name	Workflow definition name. Data type: string.
InstanceID	Workflow instance ID. Data type: string.
Status	Workflow instance status. Possible values: Running.
StartTimeUTC	Workflow instance start time. Date and time are displayed as UTC.



Column Name	Description
TaskName	The name of the workflow step which generated this task. Applies to step of type Task, Task Chain and Parallel Task. Data type: string.
TaskID	The task ID that is generated by the Pulse Engine, also known as Pulse ID. You can use this ID to find the task on the Manage Activities page in ION Desk. Data type: string.
TaskStatus	The task status as known in the Pulse Engine: NEW, ASSIGNED, UNASSIGNED, DONE. Tasks that are CANCELLED are not included in the report.
Assignee	The full name of the user to whom the task is assigned, if available. If the task is not assigned the value "N/A" is used. Data type: string.
Workflow Age(hours)	The number of hours since workflow started.
Task Age(hours)	The Task Age for tasks that are DONE is the number of hours since the task was created until the task was closed. The Task Age for tasks that are NEW, ASSIGNED, UNASSIGNED is the number of hours since the task was created until the current time. The Task Age is expressed in hours, rounded down. For example, if a task took 1 hour and 45 minutes, the report displays Task Age 1 hour.
<PARAM:parameter name from workflow model>	Columns for all workflow parameters with name as PARAM: < workflow parameter name>. The columns are sorted based on the alphabetical order of the parameter names. The columns contain parameter values in these cases: <ul style="list-style-type: none"> <li>• If a workflow parameter is used in a task that is DONE, the value of the parameter at the time the task was closed is shown</li> <li>• If a workflow parameter is not used in a task that is DONE, the constant "N/A" is shown</li> <li>• For open tasks, the value of the workflow parameter at the time the task started is shown. Values are also shown for all the workflow parameters.</li> </ul>

## Workflow instance status information

For each workflow instance, depending on its status, one or more rows are displayed as follows:

- Workflow instances that are finished have the Completed, Cancelled, or Failed status. For these instances, the report shows one line for the status at the end of the workflow. This line contains the parameter values as known at this point. The step type is "End" step.
- For workflow instances that are running, this information is displayed:
  - One line is displayed for each workflow step that is not yet completed. These steps have the WAITING status. The line contains the parameter values as they were when the step started. If a parameter is not used in a step, the value as known in the workflow is displayed.
  - The steps that are completed are not displayed in the report.
  - One line is displayed for the End step with status WAITING. This line contains the values as known at the time of the report generation.

### Formatting of the data in the report

The file is generated using Encoding UTF-8.

The "carriage return" character is used as line separator.

Dates and times are converted to UTC time and formatted as YYYY-MM-DD and YYYY-MM-DD HH:mm:ss.

Decimal values are exported using the dot as the decimal separator. No digit grouping is used. To avoid that scientific notation characters are displayed, decimal values are rounded to 16 decimal places.

Boolean values are exported as true/false.

String values that contain special characters are escaped as follows:

- Non-existing values are expressed as "N/A".
- Empty strings ("" ) are expressed as is. No quotes are displayed.
- Commas (,) are replaced by semicolons (;).
- Characters for new line (`\n` or `\r\n-CRLF`) are replaced by empty spaces.

## Downloading reports

**1** Observe the count of [Reports](#) in the **Reports** menu entry. The number of reports that are ready for you to download is displayed in this counter.

**2** Click the [Reports](#) menu entry to navigate to the Reports page.

The list of all reports you have generated is displayed. You can download the reports to your local drive, or delete them directly.

You can generate the report again by applying the same filter in the pages for managing activities or workflow instances.

## Configurations for ION Pulse

### Customizing the layout for email notifications sent by Pulse

By default, a standard template is provided for emails sent by Pulse for tasks, alerts, and notifications. This template is described in the `email_template.htm` file.

You can customize the layout of the email notifications that Pulse sends for alerts, tasks and notifications created by Event Management or Workflow. To customize, you can download the standard template from ION Desk, perform the customizations, and then upload the new templates.

- 1 Download the standard template.
  - a In ION Desk, navigate to **Configuration > ION Service > Configuration Files** and expand the **E-mail templates** section on the **Configuration Files** tab.
  - b Click **Download** to obtain the standard set of email templates from the ION installation.
- 2 Perform customizations of the email layout.
  - a Unzip the downloaded file to a local folder and perform the required customization. The procedure to perform the customization is explained later.
  - b By default, a standard template is used as described in the `email_template.htm` file. You can customize this template or you can specify a new template file called `custom_template.htm`.

Alternatively, you can specify a different template for each monitor, each workflow, or each step in a workflow in addition to the `email_template.htm` file. You can specify translated templates in a language of your choice.

You can also use an additional `.properties` file to specify different email properties.
- 3 Upload the customized email templates.
  - a Create customized email templates and ensure the changes are according to HTML standards.
  - b Once the customizations are completed, zip the files again. Ensure that the root folder where the files are stored is not added to the archive.
  - c In ION Desk, navigate to **Configuration > ION Service > Configuration Files** and expand the **E-mail templates** section on the **Configuration Files** tab.
  - d Click **Upload Updated Set of Templates** to upload the new zip file to the server.
  - e Click **Save** on the **Configure** page. The new templates are now available for email notifications from Pulse.

### Customizing email properties and templates

These properties are used by default in the email notifications sent by Pulse:

- The email notifications have as sender the email address that is configured in the email settings from the ION Service properties.
- The default Infor logo from the email templates set is used.
- Data is displayed as trees in the email body.
- The email subject is the same as the task, alert, or notification message.

Using a `.properties` file, you can specify the custom behavior for each monitor, workflow, or individual task or notification from a workflow.

You can use these custom behavior properties:

- `logo` property, to specify a different logo.
- `fromAddress` property, to specify a different sender email address.
- `DisplayTreesAsTable` property, to specify whether trees must be shown as tables for alert data and workflow structures included in tasks and notifications
- `emailSubject` property, to specify a custom email subject.

**Note:** The property names that are used in the `.properties` files are case sensitive.

To specify a different email layout for a specific monitor, or a specific workflow, create a `.htm` template file. The template file must contain the name of the monitor or workflow definition. The standard `email_template.htm` file is required for the monitors and workflows that do not have a customized template. Therefore you must create the new templates in addition to `email_template.htm` file.

The `.properties` and `.htm` files can be specified for a specific monitor, workflow, or a task or notification from a workflow as follows:

- For a monitor, create a `.properties` and a `.htm` file that contains the name of the monitor. Place these files in the root folder with email templates:
  - `monitor_properties_monitor name.properties`
  - `monitor_template_monitor name.htm`
- For a workflow, create a `.properties` and a `.htm` file that contains the name of the workflow. Place these files in the root folder with the email templates:
  - `workflow_properties_workflow name.properties`
  - `workflow_template_workflow name.htm`
- For a task or notification from a workflow, follow these steps:
  - For each task or notification to be customized, create one file with the `.properties` extension to customize the email properties and one file with the `.htm` extension to customize the email body. For the same task or notification, use the same file name for both extensions. We recommend that the file name starts with “`step_`”. The file name does not have to be the same as the task or notification name.
  - In the email templates folder, create a subfolder with the same name as the workflow definition name that generates this task or notification. Place the customized task and notification files in this subfolder.
  - Add a section in the workflow properties file. Specify the mapping between the task or notification name used in the workflow and the name of the files for the customized email templates. The task and notification names may contain spaces. In this case they must be surrounded by braces `{}`.

See [Example of customized email templates for a task and notification](#) on page 429.

**Note:**

- The name comparison for workflow and monitor names is case-sensitive.
- If a property or template is not specified at the step level for a task or notification, the workflow email templates or properties are used.

If there is no customization at the workflow level, the standard email templates and properties are used.

- If a property or template is specified incorrectly at any level, the default setting is applied, with the exception of the `fromAddress` property.  
If the `fromAddress` property contains an invalid email address, an error is logged in the `Process Pulse.log` file and the email is not sent.
- If the step name changes in the workflow, you must also update the step name in the workflow properties mapping file.

### Example of customized email templates for a task and notification

Assume the `Purchasing_Workflow` workflow has one task called “Purchase approval” and one notification called “Inform requester”. To customize the email notifications for the task and notification, you must complete these steps:

- 1 Create the `step_PurchaseApprovalTemplate.properties` and `step_PurchaseApprovalTemplate.htm` files. These files will be used as customized email template for the task.
- 2 Create the `step_UserNotificationTemplate.properties` and `step_UserNotificationTemplate.htm` files. These files will be used as customized email template for the notification.
- 3 Place these four files in the `Purchasing_Workflow` subfolder.
- 4 In the root folder for email templates, place the workflow properties file called `workflow_properties_Purchasing_Workflow.properties` and add this mapping:

```
# workflow step mapping to email templates
{Purchase approval}=step_PurchaseApprovalTemplate
{Inform requester}=step_UserNotificationTemplate
```

## Custom email properties

### logo

To use a custom logo image in the email notifications:

- Add the logo file to be used to the folder with email templates. All mime formats are supported. For example, use png, gif, or jpeg.
- If the logo file must be used for a specific task or notification email notification. Add it to the subfolder with the same name as the workflow that generates the task or notification.
- Edit the content of the applicable `.properties` files to contain this line:

```
logo=companylogo.png
```

Where `companylogo.png` is the name of the file to be used as logo.

**Note:** If the property is not specified although other properties or email template are customized, or if the logo file name is not correct. The default Infor logo is used.

### fromAddress

To use a custom email address displayed as “From:” in the email header, edit the content of the applicable `.properties` files. The files must contain this line:

```
fromAddress=name@company.com
```

Where the value `name@company.com` must be a valid email address.

**Note:**

- Depending on the email server configuration, this email address can be a verified email address. Contact your system administrator for details.
- If this property is not specified, the From address that is configured in the ION Service Properties configuration for email is used. If the `fromAddress` property is specified but it is incorrect, an email notification is not sent and an error is logged in the Pulse log file.

**DisplayTreesAsTable**

Use the `DisplayTreesAsTable` property to specify that tree structures should be displayed as tables in alert details or workflow structures from tasks and notifications. Edit the content of the applicable `properties` files to contain this line:

```
DisplayTreesAsTable=True
```

**Note:** If this property is not specified or the value is anything else than 'True'. The data from alert details and workflow structures is displayed as indented trees in the email contents.

The table to display alert details is built as follows:

- A table is created for each noun included in the alert details.
- The tree root becomes the table header. This is the noun name.
- The tree leaves become column names. The `SequenceNumber` information added to the trees is not included as a column name.
- For each component level in the tree, a new row is created in the table. The level names are not displayed in the table.
- Cell values are filled with values of the tree elements. Values of elements from parent levels are repeated on each row for each child.
- If values are missing for a leaf from a child level, the cell is left empty. If a level does not have any values for the leaves, except the `SequenceNumber`. The empty row that would be generated for this level is removed from the table.

The table to show workflow structures in tasks and notifications is built as:

- For each structure, a different table is created.
- The structure name becomes the table header.
- The names of the fields in the structure become column names.
- For each structure level, a new row in the table is created.
- Cell values are filled in with the field values from the structure.
- If the structure has fields on the root level, these values are repeated on each row in the table.
- If a level has a sublevel, a row is created for each occurrence of the sublevel in the structure. The values of the fields from the parent level are repeated on each row that is created for an occurrence of the sublevel. .

**emailSubject**

Use the `emailSubject` property to specify a custom email subject for alerts, tasks, or notifications as follows:

- For alerts that are generated by a monitor, use this property in the monitor `.properties` file:  
`emailSubject= String to specify customized email subject`  
 The customized email subject is a string that cannot contain placeholders. The email subject is constructed by appending the alert status, such as “New Alert”, and the customized subject string.
- For tasks or notifications that are generated by a workflow, use this property in the step-specific properties file. You can also use this property in the workflow `.properties` file to ensure all tasks and notifications from the same workflow have the same email subject:  
`emailSubject= String to specify customized email subject with parameter placeholders`  
 Parameter placeholders are defined as the workflow parameter name between square brackets. The parameter must be part of the task or notification content and the names must match, using case-sensitive comparison. At runtime, the parameter placeholders are replaced with the value that is given to this parameter. If a parameter placeholder is not specified correctly, the `[?]` string is displayed.  
**Note:** You cannot use the square bracket characters in the `emailSubject` and content of email templates for other purposes than to define workflow parameter placeholders.

**Note:** If this property is not specified, the task, alert, or notification message is used as the default value for the `emailSubject`.

## Customizing email templates

Using the `.htm` template files, you can customize the email templates for:

- Alerts that are generated by monitors.
- Tasks and notifications that are generated by workflows.

This table shows the elements you can use in the template file:

Placeholders	Replaced with
<code>&lt;_LOGO_IMAGE_&gt;</code>	The logo image in the file <code>logo.png</code> from the same folder.
<code>&lt;_TENANT_ID_&gt;</code>	The tenant name from where the email notification is sent.
<code>&lt;_ITEM_IMAGE_&gt;</code>	The image of the Pulse item, containing information about type, priority and status. One of the icon files that start with <code>icn_alert.png</code> , <code>icn_task.png</code> or <code>icn_notification.png</code> is used.
<code>&lt;_TYPE_TXT_&gt;</code>	The Pulse item type as text. Possible values: Alert, UserTask or Notification.
<code>&lt;_PRIORITY_TXT_&gt;</code>	The Pulse item priority as text. Possible values: Alert, High, Medium, Low, Notification.
<code>&lt;_ACTIVITY_ID_&gt;</code>	The Pulse item ID.
<code>&lt;_CREATED_TXT_&gt;</code>	The creation time of the Pulse item as text. The time is displayed according to the user timezone setting from the Infor Ming.le User Profile. If this setting cannot be retrieved, the UTC time is displayed.

Placeholders	Replaced with
<_DUEDATE_TXT_>	<p>The due date time of the Pulse item if this was configured.</p> <p>The time is displayed according to the user timezone setting from the Infor Ming.le User Profile. If this setting cannot be retrieved, the UTC time is displayed.</p> <p>This placeholder is commented out by default in the <code>email_template.htm</code> file. To see the due date in all email notifications about alerts and tasks, remove the comments around the <code>&lt;tr&gt;</code> tag that contains this placeholder. Or create custom templates with this placeholder.</p>
<_MESSAGE_TXT_>	The message of the Pulse item, in the default language.
<_URL_TO_INBOX_>	<p>A direct link to the alert, task, or notification from this email. Click the link to open the Inbox application that is hosted in the Infor Ming.le portal.</p> <p>The user must log in to Infor Ming.le and then can see the message details. If the message is open, the user can take actions. The message details are only accessible to Infor Ming.le users who have access to Inbox and are in the distribution list of the message.</p> <p>This placeholder is commented out by default in the <code>email_template.htm</code> file. To use this placeholder, remove the comments around the <code>&lt;td&gt;</code> tag that contains this placeholder. Alternatively, create custom templates with this placeholder.</p>
<_HR_ITEM_DETAILS_>	Horizontal bar separator to mark the beginning of Item Details section.
<_ITEM_DETAILS_HTML_>	The Pulse item details.
<_HR_NOTES_>	Horizontal bar separator to mark the beginning of Notes section.
<_ITEM_NOTES_HTML_>	The notes added to this Pulse item. Applies only to Alerts and Tasks.
<_ITEM_CARRY_FORWARD_NOTES_HTML_>	<p>The notes that are added in the previous tasks of the same workflow. This placeholder can be used for:</p> <ul style="list-style-type: none"> <li>• Tasks that are created by a Workflow</li> <li>• Notifications that are created by a Workflow and which have the option to include notes from previous steps.</li> </ul>
[workflow parameter name]	<p>The value of the workflow parameter name that is used in the placeholder. This applies to workflow parameters and structures. You can only use this type of placeholders for email templates for individual tasks or notifications.</p> <p>The parameters that is used in the placeholders must be part of the task or notification content definition. The parameter names must match the names used in the workflow, using case-sensitive comparison.</p> <p>If a parameter placeholder is not specified correctly, the <code>[?]</code> string is displayed.</p>



## Translating email templates

You can write the customized email templates in a language of your choice. These templates are used as default if translations for a specific language code are not available. You can specify translated custom templates for several languages, so that different users receive the same email notification in the language of their choice.

Infor Ming.le users can edit their profile information through the **User Settings** menu option. A user can select an email language in the **Email Preferences** section. If present, translated email templates that match the selected language code are used to send email notifications to the user. An email template may contain placeholders for parameters of type decimal or date and time. The values of these parameters are formatted according to the default locale that belongs to the selected language.

If one of these situations occurs, the custom email templates from the default configuration are used:

- The user does not set a preferred email language.
- No translations are found for the selected language.

## Adding translated email templates

1 Create custom templates for the `email_template.htm` or for monitors, workflows, or specific tasks and notifications as required. This is referred to as the basic structure.

2 For each language for which translations are required, add a folder with this name at the root level:  
`language_<code>`

For example, `language_en-US`.

The `<code>` in the folder names is according to the “Language Culture Name” from this overview: [https://msdn.microsoft.com/en-us/library/ee825488\(v=cs.20\).aspx](https://msdn.microsoft.com/en-us/library/ee825488(v=cs.20).aspx). Folder names are case in-sensitive.

3 Create a copy of the basic structure with the custom templates. Place this copy under the `language_<code>` folder. Translate the email templates and the `.properties` files for this language.

The properties file in the language folder can contain these translations:

- **emailSubject**. Can be a translated string; you must translate the email type in a separate file called `emailtypes.properties`.
- **logo**. A file name that can point to a file from the same folder as the properties file where it is used; you can use a different logo for each language.
- **fromAddress**. An email address. Can be different for each language.
- **DisplayTreesAsTable**. Not translatable, always **True** or **False**, but can have a different value for each language.

The email subject is composed of two parts: the email type prefix and the email subject string. To translate the email type prefixes, the `emailtypes.properties` file is provided. Translations for the predefined prefixes can be defined for each language and they are used only if they are placed in the language folder. The `emailtypes.properties` file is ignored if it is stored outside of the language folder.

It is allowed to specify translations only for a subset of the email types. If an email type is left blank in the file from the language folder, for example `NEW_ALERT=`, the email subject does not contain a prefix.

4 In the `emailtypes.properties` file you can also specify translations for the Priority property.

This property is shown only if the placeholder `<_PRIORITY_TXT_>` is included in the email template and it shows these values by default:

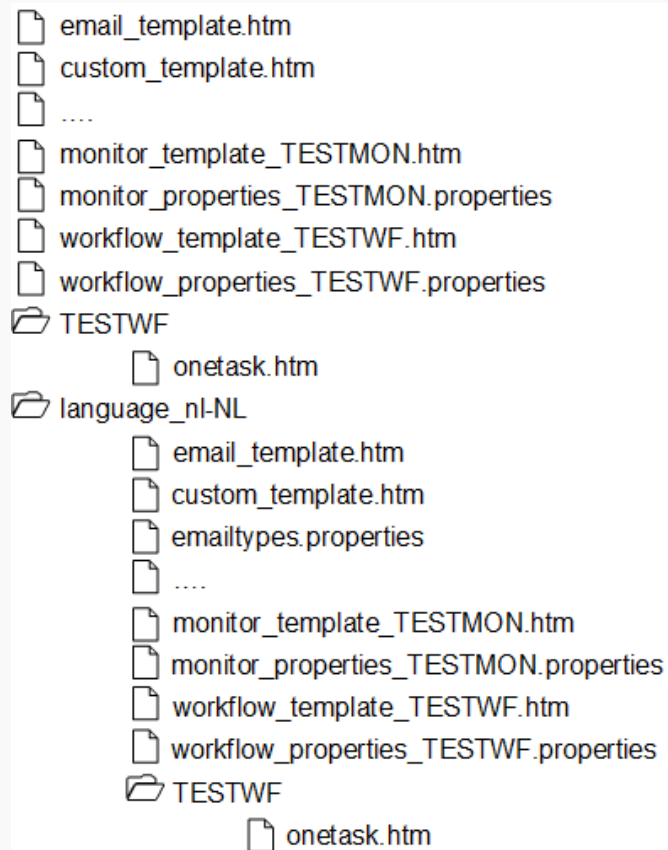
- Alert - for alerts.
- High, Medium or Low - for tasks.
- Notification - for notifications.

- 5 Archive the whole folder structure with customized email templates from the basic structure and the language folders and upload it to ION. The translated email templates are used immediately. An email that uses a template for a specific language is sent to all users who have specified that language as their email preference language.

**Note:**

- When uploading a zip file with customized and translated email templates there is no validation on the customization correctness.
- After the user changes the language settings in Infor Ming.le, it may take up to 12 hours until the change is reflected in emails. If a new set of customized email templates is uploaded meanwhile, the language settings take effect immediately.
- The `.properties` file from the language folder may not contain different mappings between workflow step names and template names. If they are present, these mappings are ignored.

This diagram shows an example of a set of customized email templates with translations:



The set consists of these elements:

- The basic structure
- One language folder: `language_nl-NL`  
Files and sub folders must have the same names as in the basic structure, but contents in these files are translated.

## Troubleshooting and logging

### ION status overview

The ION Home page serves as a dashboard that lists these statistics:

- Active data flows.
- Active connection points including the count of connection points in error state.
- Unhandled Error Messages.
- Active event monitors.
- Active alarm templates.
- Active workflows
- Active activation policies
- Active workflow schedules
- Open activities

### Viewing status of active data flows or connection points

To verify if there are any problems in processing BODs, for example because of connection point, go to the **Home** page. Check the overview status of active connection points.

To view a summary of ION connect status:

**1** Select **Connect > Active Connection Points**.

This page shows the number of active data flows and active connection points. An indication of the number of connection points in error state is also displayed.

**2** Click one of the hyperlinks to drill down to the corresponding manage page.

### Viewing the Error Messages

To verify if there are unhandled Error Messages:

**1** Select **Connect > Active Connection Points**.

This page shows a link to the **Error BODs** page with an indication of number of unhandled Error Messages specified in the bracket.

- 2 Click the hyperlink to drill down to the **Error BODs** page.  
Click the hyperlink on the number of unhandled BODs to directly go to the filtered view of list of unhandled Error Messages.

## Viewing activities

To view a summary of the status of activities:

- 1 Select **Monitors & Workflows > Activities**.  
This page shows the activities.
- 2 Click the hyperlink to drill down to the corresponding manage page.

## ION OneView

In many cases, ION OneView is a good starting point for troubleshooting. For example, it will answer questions such as:

- Was sales order 1234 published from my application?
- Where did this document go?
- Did it result in an error?

For details, see [ION OneView](#) on page 361.

## Missing data

If messages are not delivered at the place where you expect, check whether errors are reported.

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Connect > Error BODs**.  
See [Manage ION Connect](#) on page 386.
- 3 Check the status of the ION Desk Service and the location of existing messages.
- 4 On the ION Desk menu, select **Connect > Active Connection Points**.  
Check the status of the used connection points and the number of messages sent and received by each connection point.
- 5 Select **Connect > Active Data Flows**.  
Check the status of the active data flows and the number of messages processed in mappings, content-based routings and filters.  
See [Managing activities](#) on page 417.

- 6 On the ION Desk menu, select **Connect > Active Routings**.  
Verify that the routings are correct and an appropriate route is available for the messages to be delivered.
- 7 For messages that are in the outbox of an application, you can check the message status.  
If you use an Infor Application connection point, you can check the contents and status of messages in the Outbox and Inbox tables. For details on the Outbox and Inbox tables, see the *Infor ION Development Guide* .  
In the Outbox tables of an application, you can check the status of the message:
  - Status '0' means the message is waiting to be processed.
  - Status '1' means the message is processed.If the message is waiting in the Inbox of an application, it was processed successfully in ION. In that case check whether there is a problem in the receiving application.
- 8 In OneView, search for messages with event type "no route"
- 9 Use the log files.

## Incorrect data

If incorrect data was received by an application, you can use the message tracker to see in what stage of the process the problem was caused.

This way you can check whether the data was already incorrect when it was first published to ION or whether it became incorrect somewhere in the flow. You can also use this to validate whether a mapping activity, a content-based routing or filter is working properly.

For details, see the "Logging" section.

## Exception handling

The exception handling in the ION Service:

- If the connection succeeds but subsequent processing fails, this is regarded as a functional error. In that case a Confirm BOD is sent and the next message will be processed.
- If the connection fails, retrying is useful (assuming the connection was tested before activating the document flow). In that case no Confirm BOD is published. Retrying will be done automatically. In that case the message will stay in the same location (for example an Outbox or an intermediate message queue) until connection succeeds.

If a Confirm BOD is published this is handled in ION and displayed on the Error BODs page, as described in the Management chapter.

When using database connection points, the ION Service will make sure that the database transaction is handled properly. Changes to the database and updates to the involved message queue are done together. They are either both committed or both rolled back. Note that the handling of an incoming Process BOD and the publishing of the resulting Acknowledge message is synchronous. If writing the

Acknowledge message to a queue failed, the action in the database and the removal of the Process message from the incoming queue will also be rolled back.

## Receiving e-mails for exceptions

You can configure ION to send an e-mail in specific exception situations.

**1** Select **Configuration > Error reporting**.

**2** Select one of these check boxes when an e-mail must be sent:

**An Error BOD occurs:**

You want to receive an e-mail if an Error Message was created, either by one of the connection points or by the ION Service.

**A document is published having no routing defined:**

You want to receive an email if a message is published by a connection point, but there is no place to send this document to. In other words, no document flow is defined for this BOD, and no event monitor or workflow activation policy is active for this document.

**3** Specify the e-mail address of the person who must receive the e-mails. To specify several e-mail addresses, use a semicolon or a comma separator.

**Note:** The SMTP server and the 'from' e-mail address that are pre-configured by the system administrator are used when sending these e-mails.

## Receiving alerts for business process errors

You can configure ION to send an alert when business process errors occur. For example, send an error alert when a:

- Task, notification or alert was created with an empty distribution list.
- Workflow instance failed.
- A workflow schedule failed to start a workflow.
- Monitor, activation policy or alarm template reached the *Error* status.

You can specify one user account or a group name that receives all error alerts.

To configure sending alerts for business process errors:

**1** Select **Configuration > Error Reporting**.

**2** Select the **Generate alerts for business process errors** check box in the **Business Process Administrator** section.

**3** Click **Select** next to the field **Send Alert To** field to specify the distribution list of the error alerts.

**4** Select a type User or Group in the **Select Distribution Data** dialog box.

**5** Select a user or group for distribution Only one user or one group can be selected.

**6** Optionally, select the **Send an email notification** check box for the error alerts.

**7** Click **OK** to return to **Error Reporting**.

Verify that the **Send Alert To** field is filled according to your selection.

**8** Click **Save** to confirm the changes.

The alerts that are created in error situations are sent to the configured users. The alerts are visible in Infor Ming.le in the same widgets as the business alerts.

To disable sending of alerts for business process errors, go to **Error Reporting** and clear the **Generate alerts for business process errors** check box. Click **Save** to confirm the changes.

**Note:** the alerts generated for business process errors are visible in the **Monitors & Workflows > Activities** pages with Source Type: *Error* and Source: *PulseEngine*, *MonitorEngine* or *WorkflowEngine*. These alerts are archived when they are closed.

## Receiving alerts for exceptions

To receive an alert in case an Error Message occurs, you can define an event monitor.

**1** In the **Monitors** modeling page, click **Add** to create a monitor.**2** Specify this information:**Name**

Specify a name for the monitor.

**Tenant**

Select a tenant to monitor another than the default tenant. Note: in case of multiple tenants, you need to set up multiple event monitors.

**3** Click the **Application Documents** tab and add a document with the name 'Error'.**4** Click the **Attributes** tab and select the data you want to include in your alert.**5** Click the **Conditions** tab, specify a condition that always evaluates to true.**6** Click the **Rule** tab, select **Rule Type** 'Condition Only' and select your **Condition**.**7** Click the **Alert** tab and specify the message text to be included in the alert.**8** Click the **Distribution** tab and select the user(s) to whom the alert must be sent.**9** Click **Save** to save the monitor and return to the list of monitors.**10** Select the Monitor and click **Activate**.

For details on how to set up an event monitor and how to specify specific alert conditions, see [Event Management](#) on page 195.

For details on how to handle alerts, see these guides:

- *Infor Ming.le User Guide*
- *Infor Inbox User Guide*

## Chapter 14: Configuration

You can perform several configuration tasks from ION Desk.

This table shows the structure of the Configuration menu:

Page	Usage
Codes	Define codes to be used in workflow parameters.
Error Reporting	Configure how the system administrator must be informed of errors.
ION Service	View or change general properties for the ION Service.
Downloads	Download available files and review the history of downloads.

### Codes

User defined Codes are used in Workflow and Event Management configuration. In ION, Codes are predefined lists of "code - description" pairs.

In Event Management, you can use Codes to define comparison conditions with the predefined values.

In Workflow, you can use Codes to specify possible choice values for workflow parameters of the Code type. When these parameters are used as input parameters in tasks, they are displayed as a dropdown in Infor Ming.le.

#### Shared Codes

Some Codes originate from code lists that are created in another application, for example currencies and country codes from ERP LN. You cannot import and reuse these lists in ION.

### Creating codes in ION

User Defined Codes for ION are value codes to be used in workflow models and Pulse. For example: reason codes for requisition rejection from LN is a pre-defined code. This code is created in LN and must be reused in Workflow.



---

To manually create a Code in ION:

- 1 Select **Configuration > Codes**.
- 2 Click the plus icon in the **Codes** screen to add a new Code.
- 3 Specify a Code name and description.
- 4 Click the plus icon in the code details section to add a code and a description to the list.
- 5 Click **Save** to save the new code. The code is now available for usage by **Workflow Modeler** and **Pulse**.

## Validation rules

- Code names must be unique for an ION installation.
- Codes and descriptions within a code must be unique.
- Codes that are in use by workflows or monitors that are configured in this ION installation cannot be deleted. Even if the monitors and workflows are not active.
- Code details can be edited at any time:
  - New entries with new codes and descriptions can be added without restrictions.
  - Descriptions can be edited without restrictions.
  - Codes cannot be deleted if a code is in use, see bullet 3.
  - Codes can only be deleted from codes that are not in use.
- Usage of special characters:
  - Codes: can be limited to a predefined list of characters.
  - Descriptions: special characters and multibyte codes are allowed.
- Upon editing, changes in codes are immediately available to Event Management, Workflow, and Pulse.

## Importing codes

- 1 Select **Configuration > Codes**.
- 2 Click **Import**.  
A dialog box is displayed to browse the file system.
- 3 Select an xml file from the file system and click **OK**.  
All codes in the file are added to the list of ION Codes.
- 4 If the file does not have the correct format, one of these error messages is displayed:
  - `This file does not contain ION Codes`
  - `Invalid import contact your system administrator for details`Only files that are exported from an ION installation can be used.
- 5 Click **OK**. Nothing is imported.
- 6 If the file contains code names that are the same as the codes that are already in the system, a warning is displayed. You are asked to select one of these actions:

- **Skip**  
Duplicate codes are skipped, and logged as warnings in the result dialog box.
- **Rename**  
Duplicate codes are renamed using '\_' and a number, so the names are unique.

## Exporting codes to XML

- 1 Select **Configuration > Codes**.
- 2 Select one or more codes and click **Export**.
- 3 A dialog box opens to select a location on the file system.
- 4 To save codes to a file, browse to a location on the file system and use the default name `ION_Codes_<date>.xml`. The variable `<date>` is the current date of the system.

Codes Page toolbar:

The icons above the Codes table are:

- Discard Changes
- Refresh
- Add
- Save
- Delete
- Import
- Export

The icons above the Code details table are:

- Discard Changes
- Add
- Save

- 5 Click **OK**.

## Configuring the ION Service

The ION Service uses preconfigured property settings. After installing ION, all required properties are set using a default value, or using input from the installation. Sometimes you can adapt the settings after installation.

To open the page where you can view or update the settings, select **Configuration > ION Service**.

On this page, the **Properties** tab consists of changing values of the Pulse engine property to send email. The **Configuration Files** tab provides the option to upload or download the email templates for the email notifications sent by Pulse for the current tenant.

## Properties tab

The available settings for ION Connect in the **Properties** tab and their description are listed in this table:

Group	Property	Description
Pulse engine	Send E-mail	This property indicates whether emails must be sent from Pulse, for example if a task or alert is created.
	Auto Assign Activities	This property indicates whether alerts and tasks are automatically assigned to the user if there is only one user in the distribution list. The default value is <b>False</b> . When this property is set to <b>True</b> an email is sent to the user, for the assignment event.
	No Assignment if Out of Office	This property indicates that Infor Ming.le users cannot assign a task or alert to a user who uses the <b>Out of office</b> reply. Switch this property on, to avoid assigning items to a user with the status <b>out of office</b> . Switch this property off, to allow assigning items to a user with the status <b>out of office</b> . The user can handle the item on return. The default status is <b>False</b> .
	From address	The sender email address to be used for sending emails from Pulse. Emails are sent only if the <b>Send E-mail</b> property is enabled. If this property is not specified, the default value that is specified during the ION installation is used. If a <b>From address</b> is specified in a customized email template, the customized <b>From address</b> is used.
Event Management engine	Logging triggers	This property indicates whether trigger data is logged. You can set the property to <b>True</b> or <b>False</b> . If <b>Logging triggers</b> is set to <b>False</b> , logging of triggers is disabled, and a notification is displayed above the triggers list on the <b>Show Triggers</b> page.

## Configuration Files tab

On the **Configuration Files** tab you can find these sections:

- **Certificate Store**

When secured external URLs are contacted from ION, for example: from the web service connection point or from an Infor Cloud connection point, ION will automatically retrieve the public key to connect to the service when you load the WSDL and will verify if the certificate is trusted in ION. If the certificate is already trusted by a CA, such as Verisign, then ION allows access to the URL. If the certificate is self-signed or is not trusted, ION alerts the user from where the trigger to contact the external URL was made. This can be when clicking **Test** in an Infor Cloud connection point or clicking **Load WSDL** of the web service connection point.

To check the authenticity of the certificate you can choose to trust the certificate or the issuer or both. Accordingly you can choose whether to download the certificate of the server you are calling or the issuer.

Note that the only SSL server certificates which are supported by ION are X.509.v3 certificates.

After you downloaded the certificate, inspected it and are satisfied with the authenticity of the certificate you can import it into ION's trust store:

- Click the **Import Certificate**
- A window is displayed. Browse to the location of your certificate and select it.
- Click **Open**.
- The certificate is now imported into ION and you can continue working with the external URL

- **E-mail Templates**

Pulse enginePulse uses a pre-defined set of templates for these emails. You can customize these templates. The customized templates have effect for all emails sent by Pulse for the current tenant. can send email notifications for tasks, alerts and notifications.

- **Download Scripts / IMS API Documentation**

#### **Download Scripts to create I/O box**

To connect to ION using the Infor Enterprise Connector Application connection point, create in-box and outbox tables in your database. To download the scripts to generate the I/O box:

- 1 Click **Download Scripts to create I/O box**.
- 2 Download the zip file to a folder of your choice.
- 3 Extract the zip file to a folder of your choice.

This zip file contains two folders: 1.0 and 3.0.

The 1.0 folder contains the scripts to create a base I/O box for the standard database vendors that ION is tested with. This includes: MS SQL server, Oracle, DB2, DB2400, and MySQL.

The 3.0 folder contains the scripts required to prepare your I/O box to support multiple logical IDs sharing the same I/O box. The v3.0 scripts must be run on top of a v1.0 I/O box.

For Unicode-compatible BOD header fields, you can use the scripts available in the `Unicode` folder. We recommend that you use Unicode.

#### **IMS API Documentation**

You can build your own IMS integration with ION. You can use either the Infor Enterprise Connector, Infor Application Connection Point (IMS) or IMS through ION API connection point. In all cases you require the API documentation for the IMS interface that you must implement at your application side. Click **Download IMS API Documentation** to download this swagger based documentation.

See the *Infor ION Development Guide*.

## Key Management tab

On the **Key Management** tab you can manage PGP keys with these options:

- 1 Generate new Public/Private key pair.
- 2 Export public key for selected PGP key.
- 3 Set PGP key to be used for signature.
- 4 Deactivate/reactivate PGP key.
- 5 Delete deactivated PGP key.

## Downloads

On the Downloads page you can find and download features that are offered for different ION and Infor products.

To access the Downloads page, expand the ION Desk menu and select **Configuration > Downloads**. The Downloads page contains these two tabs:

- **Files**
- **History**

On the **Files** tab, each product has a section with the available features and their current version for download.

Currently, the Data Lake product with the Compass JDBC Driver feature is available for download.

For more details on Compass JDBC Driver, see the *Infor ION Development Guide*.

Each download activity is registered in a history log for auditing and acknowledgment purposes. To access the history logs, select the **History** tab on the **Downloads** page.

On the **History** tab, the latest downloads are shown in a data grid that you can sort, filter, and export.

These columns are available for the logs:

- Product – Product name of the feature
- Feature – Feature name
- Release Version – version that was downloaded
- Downloaded By – users that made the download
- Downloaded On – date and time when download on

## Downloading a feature asset

- 1 Select **Downloads**.
- 2 Find the product name section, for example Data Lake.
- 3 Click the feature name button, for example: **Compass JDBC Driver**.
- 4 Review any applicable “Terms & Requirements” that is presented before proceeding with the download.
- 5 If you agree with the terms provided, select the acknowledgment check box and click **Download**.
- 6 Save the file locally.

## Exporting the complete history log

- 1 On the **Downloads** page, click the **History** tab.
- 2 Click the **Export to CSV** icon.
- 3 Save the file locally.

## Chapter 15: Data catalog

The Data Catalog is an application that stores metadata about data objects that are used within the organization.

This table shows the structure of the Data Catalog menu:

Page	Usage
Object Schemas	Manage object schemas for the current tenant.
Schema Extensions	Manage custom user area extensions for the standard schemas, used for the current tenant.
Locale Selections	Manage the supported locales for the current tenant. This is used by applications sending localized data in their data objects.

This information is stored in the Data Catalog:

- The schema definition.
- The user or application to last update the object metadata.
- When the last update took place.

### Object schemas

On the **Object Schemas** page you can upload custom objects of type ANY, BOD, DSV, or JSON. They can only be used within the current tenant environment. The page shows a list of objects, including those in a standard library, that have already been uploaded for the current tenant.

To access the **Object Schemas** page, you require one of these ION Desk permissions:

- View: You can only view, refresh, and export the objects.
- All: You can perform all operations: import, export, delete.

To view the **Object Schemas** page, select **Data Catalog > Object Schemas**.

### Tiles overview

By default, tiles are used to show all object schemas that are registered with the Data Catalog.

Each tile shows information about an object schema. Among other things, this information is displayed:

- Object name, or noun name for BODs
- Type
- Last update timestamp
- Last update user
- The library to which the object schema belongs



For details about each of these items, see [Grid overview](#) on page 449.

The tiles are sorted from the most recently created or updated. The first position is reserved for the **Add** tile, which is the only tile that does not represent an object, but an action. This tile is used to add an object of type ANY to the Data Catalog. You can use the **Import** icon in the toolbar to import objects of other types.

## Tiles toolbar






Use the toolbar to import new object schemas to the Data Catalog.

This table shows the actions that are available from each tile

Name	Type	Description
Delete		Delete the item. This action is not available for the ANY object called "AnyDocument" or for Standard BODs.
Export		Export the object to a file. This action is not available for Standard BODs

## Single-select mode







You can only work with one object schema at a time by default. To select multiple objects, click Select in the top button toolbar, next to Import. This table describes the toolbar in single-select mode:

Name	Type	Description
Tiles view		Show objects as tiles.
Grid view		Show objects as a grid.
Refresh		Refresh the data on the page.
Import		Import one or more custom objects.
Select		A switch button is used to enable multi-select mode for bulk actions, such as exporting several definitions. When clicked, the toolbar is updated to show the actions applicable to the selected item(s).



## Multi-select mode

This table describes the toolbar in multi-select mode:

Name	Type	Description
Tiles view		Show objects as tiles.
Grid view		Show objects as a grid.
Delete		Delete the selected object(s). This action is not available for the "AnyDocument" object or for any standard BODs.
Export		Export the select object(s) to a file. This action is not available for standard BODs.
Cancel Select		A switch button is used to exit multi-select mode. When clicked, the button bar changes according to the table mentioned earlier in <a href="#">Single-select mode</a> on page 448. Any selected objects are deselected as part of this action.
Select All		Select all objects in the tile view that match the used filters.

## Grid overview

Grid view is another way to view the list of object schemas that exist in the Data Catalog.

The information that is displayed in the grid is also displayed in the tiles.

This information is displayed:

### Details

To view an object's metadata in the Data Catalog, click the drill-down icon.

### Name

The name of the object as it was specified upon upload. Object names are unique within a tenant and must be different than any other object's name, including those located in another library. Objects of type BOD represent nouns; the name of the noun is displayed as the object name.

**Note:** The object named "AnyDocument" is used to illustrate the structure of the import/export file and cannot be deleted.

### Type

The type of the object.

### Subtype

The subtype of the object. Objects of type DSV and JSON are assigned a subtype at the time that the custom object is imported.

Depending on what the separator character in the schema is, DSV objects are assigned one of these subtypes:

- CSV - comma-separated values

- TSV - tab-separated values
- PSV - pipe-separated values
- Other

For JSON objects, if the schema specifies that the object is using streaming JSON, the subtype is “Newline-delimited”. For objects that do not use streaming JSON, the subtype is “Conventional”.

### Library

The library to which the object schema belongs, for example InforOAGIS. All custom objects are considered part of the “Custom” library.

### Last Updated by

The UPN identifier of the user who performed the last upload for this object.

### Last Updated on

The date and time when this object was uploaded.

## Filter overview

- 1 To search for a specific object, or set of objects, in the Data Catalog use the filter panel on the left side of the page.
  - a Use the search box at the top of the panel to filter by object name. By default, objects that contain the keyword somewhere in their name are displayed on the page. A drop-down filter with additional options is available to filter for objects whose name starts with, ends with, or equals the specified keyword.
  - b Expand the **Library** section to filter by object library. Expand the “Standard” option to select a specific standard library that is uploaded for the tenant.
  - c Expand the **Object Type** section to filter by object type and subtype.
  - d Expand the **Last updated on** to filter by the date and time the object was last updated.

Filter options	Description
None	No filter for "last updated on" is applied. The default selection.
Today	Retrieves the objects that are updated since midnight on the current day.
Last Week	Retrieves the objects that are updated in the past 7 days.
Time Range	Use the date fields to specify a time range to use when filtering for objects. This option retrieves the objects that are updated between the “Since” date and the “Until” date.

- e Expand the **Last updated by** to filter by the last update user.
- 2 Click **Search** to run the filter.
  - 3 To reset the filter, click **Reset**.



When a filter is applied, dismissible tags are shown at the top of the page. Each selection made in the filter panel is displayed in separate tags. To remove a filter selection, click the **x** icon in the tag. After removing a tag, the filter runs again to find all objects matching the updated filter criteria. To clear the applied filter for all selections, click the **Clear All** tag.

## Schema generation wizard

Use the schema generation wizard to create a new object schema.

The wizard takes you through the correct steps to create a valid schema for your data object. Schemas can be created for JSON, DSV, and ANY objects. BODs are not yet supported.

This table shows the flow of the schema generation wizard:

Step	Page	Information
1	Create Schema	Select which method to use to create the new schema. You can use these options: <ul style="list-style-type: none"> <li>• Generate a schema from sample data.</li> <li>• Build manually</li> </ul>
2	Object Definition	Specify basic information for your object schema, such as name, description, and type. For DSV objects, define the dialect information.
3	Property Definition	Define each of the properties that are to be included in the schema.
4	Additional Metadata	Define additional metadata for your object. Select which properties are the identifier, variation, timestamp, and delete indicator.
5	Review	Review the generated schema and additional metadata properties.

## Generating a schema from sample data

- 1 Select **Data Catalog > Object Schemas**.
- 2 In tiles view, click the **Add** tile. In grid view, click the **Add** icon above the grid.
- 3 The **Generate from sample data** option is selected by default. Drag and Drop, or click **Select Files**, to upload one or more sample data files to generate a schema. All selected files must be of the same object type and subtype.

**Note:** JSON sample files must use the extension `.json` to be properly inferred.
- 4 Click **NEXT** to specify the object information. See [Object definition](#) on page 452

---

## Object definition

After clicking **NEXT** on the **Create Schema** page, the **Object Definition** page is displayed.

1 Specify this information:

### Object Name

The name of your object.

This value is inferred based on the name of the sample file provided on the previous step. This field is required and can be changed.

Object names cannot exceed 100 characters. These characters are allowed:

- Letters a-z and A-Z
- Numbers 0-9
- These special characters: underscore (\_), period (.), and hyphen (-)

### Description

Optionally, provide a description for the object.

### Object Type

The type of object schema you are creating.

This value is inferred based on the sample file and cannot be changed.

### Subtype

The subtype for the object schema.

This value is inferred based on the sample file and cannot be changed.

The next fields apply to objects of type DSV. They are hidden when any other type is selected.

### Field Separator

The character used to separate field values in the data object.

This value is inferred from the sample file and cannot be changed.

### Number of Header Rows

The number of header rows that exist in the data object. By default, the value is set to 1.

### Column Header Line

The line number that contains the column headers for the object. By default, the value is set to 1.

### Header rows included in sample data

Indicates that header rows are included in the sample file that is provided.

This check box is selected by default. If the sample file does not include header rows, clear this check box.

**Note:** We recommend that the sample data includes a header row so that the schema generator can properly infer the names of the properties in the data object.

### Enclosing Character

The character that indicates the start and end of a value. By default, **None** is selected. If any of the values in your data object use an enclosing character, it must be specified here.

### Specify Enclosing Character

This value is populated based on the selection that is made for Enclosing Character. If **Other** was selected, a value must be provided.

- 2 Click **NEXT** to proceed with the **Property Definitions** page.  
See [Property definition](#) on page 453.

## Property definition

After clicking **NEXT** on the **Object Definition** page the **Property Definitions** page is displayed. The values are inferred from the sample data, except for the column **Required**.

These fields are displayed:

### Property

The name of the property to include in the object schema.

### Data Type

The data type for the property. If the data type is Object, a drill-down icon is displayed in this column. Click this icon to view and edit its child properties.

**Note:** There are some limitations for inferring data types. Depending on the values in the sample data, `boolean` and custom `datetime` properties are inferred as a number or a string. The data type and any other associated information must be updated or added manually.

### Required

Applies to JSON objects only.

Indicates which properties are required to exist in the data object. By default, none of the properties are marked as required.

### Position

The order in which the properties should be displayed to the end user. By default, the position values reflects the order in which the properties are listed in the sample data file.

To change the inferred data, or to include additional validation keywords:

- 1 Select a row in the data grid. On the right panel, this list of fields is displayed.

### Property

Change the name of the property. A property name is required.

### Title

Provide a friendly name for your property.

### Description

Provide a description of the property.

### Position

Change the position value of the property. Any other property positions that are affected by this change is automatically adjusted.

**Data Type**

A drop-down list containing valid data types for a property. Use this to change the inferred data type. Options are Array, Boolean, Date, Date Time, Integer, Number, Object, String, and Time.

**Required**

Select the check box to mark the property as required.

**Value Restriction**

If the property should be defined as a constant, enumeration, or pattern, use the drop-down list to select one of these options.

- **None**  
Default selection.
- **Constant**  
A single value that the property must have in the data object.
- **Enumeration**  
A list of unique values that are defined for the property. The value for this property in the data object must match one of the defined values exactly.
- **Pattern**  
Applies to String properties only.  
A regular expression that the value in the data object must validate against.

**Constant**

Visible only when **Constant** is selected as the Value Restriction.

Specify the value that the property in the data object must have.

**Enumerations**

Visible only when **Enumeration** is selected as the Value Restriction.

Use the list builder to define the list of valid values for the property. Each item in the list must be unique.

To add a value to the list:

- a** Click the **Add** icon.
- b** Click inside the row that is displayed, to specify a value.
- c** Repeat for as many values as required.

To remove a value from this list:

- a** Select the value to remove.
- b** Click the **Delete** icon.

**Pattern**

Visible only when **Pattern** is selected as the Value Restriction.

Specify a regular expression that the value must validate against.

If the selected Data Type is **Array**, these fields are displayed:

### Array Items Data Type

A drop-down list that contains supported data types for Array properties. Options are Any, Integer, Number, Object, and String. Type **Any** does not restrict the values to be a specific data type; all are valid.

If **Object** is selected as the Array Items Data Type, a drill-down icon is displayed in the data type column in the data grid. Click this icon to view and edit the array object's properties.

This value is inferred from the sample file.

### Minimum Items

The minimum number of items that must be included in the array.

### Maximum Items

The maximum number of items that can be included in the array.

If the selected Data Type is **Object**, these fields are displayed:

### Minimum Properties

The minimum number of properties that must be included in the object property.

### Maximum Properties

The maximum number of properties that can be included in the object property.

If the selected Data Type is **Integer** or **Number**, these fields are displayed:

### Minimum

Specify the lowest number that is allowed for the property value.

If this value should be exclusive, select the **Exclusive Minimum** check box.

### Maximum

Specify the highest number that is allowed for the property value.

This value is inferred based on the biggest number found for this property in the sample data.

If this value should be exclusive, select the **Exclusive Maximum** check box.

### Multiple Of

Use this field to specify a number that the property value in the data object must be a multiple of.

Tip: This field can be used to specify the number of digits allowed after the decimal in the property value.

For example, in these situations:

- When two decimal values are allowed, specify 0.01 as the Multiple Of value.
- When three decimal values are allowed, specify 0.001 as the Multiple Of value.

The number of digits following the decimal in the **Multiple Of** value is the number of digits allowed in the data object property. The value for **Multiple Of** must end in 1 for desired results.

If the selected Data Type is **string**, these fields are displayed:

### Minimum Length

Specify the minimum number of characters that the property value must have.

**Maximum Length**

Specify the maximum number of characters that is allowed for the property value.

This value is inferred based on the longest string value found for this property in the sample data.

**Format**

A drop-down list that contains the standard Formats for string properties. Use the drop-down to select the format the string value must validate against.

If the selected Data Type is **Date**, **Time**, or **Date Time**, these fields are visible:

**Format**

Select the format that the property value must have in the data object. The drop-down list contains supported datetime formats. For standard dates, times, and datetimes, this value is inferred, and Standard is selected.

This table describes the different available formats for a selection that is based on the selected data type for the property:

Data Type	Format	Descriptions
Date Time	Standard	Standard datetime format according to ISO 8601. Example: 2019-12-01T09:15:00Z
	American	Standard American datetime format. Example: 12/1/2019 9:15:00 AM
	Epoch Milliseconds	Epoch Milliseconds datetime format. Example: 1575191700000
	Three Character Month	Datetime format using the three-character abbreviation for the month. Example: 01-DEC-2019 09:15:00
	Other	Any other datetime format that is used by the property in the data object.



Data Type	Format	Descriptions
Date	Standard	Standard date format according to ISO 8601. Example: 2019-12-01
	American	Standard American date format. Example: 12/1/2019
	Basic Date	Basic date format according to ISO8601 standards. Example: 20191201
	Other	Any other date format that is used by the property in the data object.
Time	Standard	Standard time format according to ISO 8601. Example: 09:15:00
	Other	Any other time format that is used by the property in the data object.

### Custom format

If **other** is selected in the **Format** drop-down list, you can specify the custom format the property uses in the data object. This field is hidden if any other format is selected.

For more information regarding datetime properties, see “Using datetime formats” in the *Infor ION Development Guide*.

- 2 Click **NEXT** to proceed with the **Additional Metadata** page.

## Additional metadata

After clicking **NEXT** in the **Property Definitions** page you can define the additional metadata.

- 1 Specify this information:

### Identifier Paths

Click the icon inside the field to launch the lookup widget. Select from the list one or more properties that make up the unique identifier for your data object.

### Variation Path

Click the icon inside the field to launch the lookup widget. Select from the list which property contains the variation number for your data object.

**Timestamp Path**

Click the icon inside the field to launch the lookup widget. Select from the list which property contains the update timestamp for your data object.

**Delete Indicator Path**

Click the icon inside the field to launch the lookup widget. Select from the list which property is used as the delete indicator.

**Delete Indicator Value**

Specify the value to indicate the object is marked as deleted.

This field is required when a **Delete Indicator Path** is selected. When **Delete Indicator Path** is blank, this field is disabled.

**Additional Properties**

Specify any other information to include with your object schema. An upload option is available to select a file, or you can click inside the text area to manually add text. Valid JSON must be provided.

For more information about these properties, see “Defining additional object metadata properties” in the *Infor ION Development Guide*.

- 2 Click **NEXT**, to review the generated schema and additional metadata properties.  
If any edits are required, click **PREVIOUS** to return to any of the previous pages to make changes. Alternatively, click the page title in the "tick" widget at the top of the wizard.
- 3 Click **FINISH** to save the new schema.  
A validation check is performed. If any errors occur, a message is displayed. All errors must be solved before the schema can be saved.  
If all validation rules are met, the wizard closes. The **Object Schemas** page reloads with your new object that is listed first.

## Building a schema manually

- 1 Select **Data Catalog > Object Schemas**.
- 2 In tiles view, click the **Add** tile. In grid view, click the **Add** icon above the grid.
- 3 Select the **Build manually** option.
- 4 Click **NEXT** to specify the object information. See [Object definition](#) on page 458.

## Object definition

After clicking **NEXT** on the **Create Schema** page, the **Object Definition** page is displayed.

- 1 Specify this information:

**Object Name**

The name of your object. This field is required.

Object names cannot exceed 100 characters. These characters are allowed:

- Letters a-z and A-Z
- Numbers 0-9
- These special characters: underscore (\_), period (.), and hyphen (-)

**Description**

Optionally, provide a description for the object. If **ANY** is the selected object type, this field is disabled.

**Object Type**

The type of object schema you are creating. By default, **ANY** is selected.

**Subtype**

The subtype for the object schema, if it applies. When **ANY** is the selected object type, this field is disabled.

The next fields apply to objects of type DSV. They are disabled when any other type is selected.

**Field Separator**

The character used to separate field values in the data object. This value is populated based on the selected subtype. If **Other** is selected as the subtype, you must provide a separator for the data object. The provided value must be a single character.

**Number of Header Rows**

The number of header rows that exist in the data object. When not specified the default value is set to 0.

**Column Header Line**

The line number that contains the column headers for the object.

**Header rows included in sample data**

Indicates that header rows are included in the sample file that is provided.

This check box is not used and therefore disabled when building a schema manually.

**Enclosing Character**

The character that indicates the start and end of a value. By default, **None** is selected. If any of the values in your data object use an enclosing character, it must be specified here.

**Specify Enclosing Character**

This value is populated based on the selection that is made for Enclosing Character. If **Other** was selected, a value must be provided.

- 2 If creating a new ANY object, click **Finish**. Otherwise click **NEXT** to proceed with the **Property Definitions** page.

## Property definition

After clicking **NEXT** on the **Object Definition** page the **Property Definitions** page is displayed.

These fields are displayed:

**Property**

The name of the property to include in the object schema.

**Data Type**

The data type for the property. By default, the data type **String** is selected.

If the data type is **Object**, a drill-down icon is displayed in this column. Click this icon to view and edit its child properties.

**Required**

Applies to JSON objects only.

Indicates which properties are required to exist in the data object.

**Position**

The order in which the properties should be displayed to the end user.

A single empty row in the data grid exists and is selected to define the first property. By default, String is the data type and the position value is 1.

You can add the property name, change the data type, or include additional constraints for each property.

- 1 Select the row in the data grid. On the right panel, this list of fields is displayed.

**Property**

Change the name of the property. A property name is required.

**Title**

Provide a friendly name for your property.

**Description**

Provide a description of the property.

**Position**

Use this field to change the assigned position.

Change the position value of the property. Any other property positions that are affected by this change is automatically adjusted.

**Data Type**

A drop-down list containing valid data types for a property. Options are Array, Boolean, Date, Date Time, Integer, Number, Object, String, and Time.

**Required**

Select the check box to mark the property as required.

**Value Restriction**

If the property should be defined as a constant, enumeration, or pattern, use the drop-down list to select one of these options:

- None  
Default selection.
- Constant

A single value that the property must have in the data object.

- Enumeration

A list of unique values that are defined for the property. The value for this property in the data object must match one of the defined values exactly.

- Pattern

Applies to String properties only.

A regular expression that the value in the data object must validate against.

### Constant

Visible only if **Constant** is selected as the Value Restriction.

Specify the value that the property in the data object must have.

### Enumerations

Visible only if **Enumeration** is selected as the Value Restriction.

Use the list builder to define the list of valid values for the property. Each item in the list must be unique.

To add a value to the list:

- a Click the **Add** icon.
- b Click inside the row that is displayed, to specify a value.
- c Repeat for as many values as required.

To remove a value from this list:

- a Select the value to remove.
- b Click the **Delete** icon

### Pattern

Visible only if **Pattern** is selected as the Value Restriction.

Specify a regular expression that the value must validate against.

If the selected Data Type is **Array**, these fields are displayed:

#### Array Items Data Type

A drop-down list that contains supported data types for Array properties. Options are Any, Integer, Number, Object, and String. Type **Any** does not restrict the values to be a specific data type; all are valid.

If **Object** is selected as the Array Items Data Type, a drill-down icon is displayed in the data type column in the data grid. Click this icon to view and edit the array object's properties.

#### Minimum Items

The minimum number of items that must be included in the array.

#### Maximum Items

The maximum number of items that can be included in the array.

If the selected Data Type is **Object**, these fields are displayed:

#### Minimum Properties

The minimum number of properties that must be included in the object property.

**Maximum Properties**

The maximum number of properties that can be included in the object property.

If the selected Data Type is **Integer** or **Number**, these fields are displayed:

**Minimum**

Specify the lowest number that is allowed for the property value.

If this value should be exclusive, select the **Exclusive Minimum** check box.

**Maximum**

Specify the highest number that is allowed for the property value.

If this value should be exclusive, select the **Exclusive Maximum** check box.

**Multiple Of**

Specify a number that the property value in the data object must be a multiple of.

Tip: This field can be used to specify the number of digits that are allowed after the decimal in the property value.

For example, in these situations:

- If two decimal values are allowed, specify 0.01 as the Multiple Of value.
- If three decimal values are allowed, specify 0.001 as the Multiple Of value.

The number of digits following the decimal in the **Multiple Of** value is the number of digits allowed in the data object property. The value for **Multiple Of** must end in 1 for desired results.

If the selected Data Type is **String**, these fields are displayed:

**Minimum Length**

Specify the minimum number of characters that the property value must have.

**Maximum Length**

Specify the maximum number of characters for the property value.

**Format**

A drop-down list that contains the standard Formats for string properties. Use the drop-down to select the format the string value must validate against.

If the selected Data Type is **Date**, **Time**, or **Date Time**, these fields are displayed:

**Format**

Select the format that the property value must have in the data object. The drop-down list contains supported datetime formats.

This table describes the different available formats for a selection that is based on the selected data type for the property:

Data Type	Format	Descriptions
Date Time	Standard	Standard datetime format according to ISO 8601. Example: 2019-12-01T09:15:00Z
	American	Standard American datetime format. Example: 12/1/2019 9:15:00 AM
	Epoch Milliseconds	Epoch Milliseconds datetime format. Example: 1575191700000
	Three Character Month	Datetime format using the three-character abbreviation for the month. Example: 01-DEC-2019 09:15:00
	Other	Any other datetime format that is used by the property in the data object.
Date	Standard	Standard date format according to ISO 8601. Example: 2019-12-01
	American	Standard American date format. Example: 12/1/2019
	Basic Date	Basic date format according to ISO8601 standards. Example: 20191201
	Other	Any other date format that is used by the property in the data object.
Time	Standard	Standard time format according to ISO 8601. Example: 09:15:00
	Other	Any other time format that is used by the property in the data object.

### Custom format

If **Other** is selected in the **Format** drop-down list, specify the custom format the property uses in the data object. This field is hidden if any other format is selected.

For more information regarding datetime properties, see “Using datetime formats” in the *Infor ION Development Guide*.

- 2 Click the **Add** icon to add another property to the schema.
- 3 Repeat steps 1 and 2 until all properties that are required for the schema are added to the data grid.  
To remove any properties from the list, select the row to remove and click the **Delete** icon.
- 4 Click **NEXT** to proceed with the **Additional Metadata** page.

## Additional metadata

After clicking **NEXT** in the **Property Definitions** page you can define the additional metadata.

- 1 Specify this information:

### Identifier Paths

Click the icon inside the field to launch the lookup widget. Select from the list one or more properties that make up the unique identifier for your data object.

### Variation Path

Click the icon inside the field to launch the lookup widget. Select from the list which property contains the variation number for your data object.

### Timestamp Path

Click the icon inside the field to launch the lookup widget. Select from the list which property contains the update timestamp for your data object.

### Delete Indicator Path

Click the icon inside the field to launch the lookup widget. Select from the list which property is used as the delete indicator.

### Delete Indicator Value

Specify the value to indicate the object is marked as deleted.

This field is required when a **Delete Indicator Path** is selected. When **Delete Indicator Path** is blank, this field is disabled.

### Additional Properties

Specify any other information to include with your object schema. An upload option is available to select a file, or you can click inside the text area to manually add text. Valid JSON must be provided.

For more information about these properties, see “Defining additional object metadata properties” in the *Infor ION Development Guide*.

- 2 Click **NEXT**, to review the generated schema and additional metadata properties.



---

If any edits are required, click **PREVIOUS** to return to any of the previous pages to make changes. Alternatively, click the page title in the "tick" widget at the top of the wizard.

**3** Click **FINISH** to save the new schema.

A validation check is performed. If any errors occur, a message is displayed. All errors must be solved before the schema can be saved.

If all validation rules are met, the wizard closes. The **Object Schemas** page reloads with your new object that is listed first.

## Adding objects of type ANY

You can manually add objects of type ANY by specifying only their name.

**1** Select **Data Catalog > Object Schemas**.

**2** In tiles view click the **Add** tile. In grid view, click the **Add** icon above the grid.

**3** Select **Build manually** and click **Next**.

**4** Specify the name for your object in the **Object Name** field and click **Finish**. A new object with the specified name and type ANY is added to the list.

In object names, only these characters are allowed:

- Letters a-z and A-Z
- Numbers 0-9
- These special characters: underscore (\_), period (.), and hyphen (-).

The maximum object name length is 100 characters. When importing object definitions, any standard letter from any language is allowed.

## Importing objects

**1** Select **Data Catalog > Object Schemas**.

**2** Click **Import** to upload a set of custom objects.

**3** Browse to the local file archive that contains the object definition and click **OK** to upload this file.

The archive can contain definitions for one or more custom objects. For details about how to prepare a custom object definition, see *Infor ION Development Guide*.

**4** After the upload is complete a message is displayed, showing the result of the import.

If there are validation errors, you must correct the definitions and try to import again.

If all validation rules are met, the objects included in the file archive are imported and added to the list. The list of custom objects is automatically refreshed after an import. You can see the new objects and the existing objects are updated. Your user identifier is displayed at the bottom of the new or updated object's tile or in the **Last Updated by** column.

## Zip file structure for import

A ZIP-file is used during the import and export. This table shows the format of the ZIP file:

Folder Structure	Explanation
ANY	FOLDER: This folder contains all objects of type ANY.
+-- ObjectName1	FOLDER: This folder contains all files related to this object. It has the name of the object.
+---- ObjectName1.xml	FILE: This file has the same name as the object, followed by an "XML" extension. For objects of type ANY this file is empty.
+-- ...	Other object of type ANY.
BOD	FOLDER: This folder contains all objects of type BOD.
+-- NounA	FOLDER: this folder contains all files related to this noun/object. It has the name of the noun/object.
+---- NounA.xsd	FILE: Noun/object schema file. If this schema file contains references to other schema files, these must also reside in this folder. This file must always be present and has the name of the noun/object.
+---- NounA.xml	FILE: Noun properties metadata file. This file must always be present and must at least contain the IDXPath element.
+---- ...	FILE: Any schema files that are referenced by the NounA.xsd. <b>Note:</b> No sub folder structure may be present here!
+-- ...	Other object of type BOD.
JSON	FOLDER: This folder contains all objects of type JSON.
+-- Object1	FOLDER: This folder contains all files related to this object. It has the name of the object.
+---- Object1.schema.json	FILE: The JSON schema that describes this object. It has the name of the object and a ".schema.json" extension. The file must use UTF-8 encoding and must be a valid schema file according to JSON schema definition, version 6.

Folder Structure	Explanation
+---- Object1.properties.json	FILE: A JSON file that describes additional properties for this object. It has the name of the object and a ".properties.json" extension. The file must use UTF-8 encoding and must be a valid JSON file according to the schema for the properties file. For details on this schema, see the <i>Infor ION Development Guide</i> .
+-- ...	Other objects of type JSON. Only one JSON schema file for each object must be used.
DSV	FOLDER: This folder contains all objects of type DSV (Delimiter-Separated Values).
+-- Object1	FOLDER: This folder contains all files related to this object. It has the name of the object.
+---- Object1.schema.json	FILE: The DSV schema that describes this object. It has the name of the object and a ".schema.json" extension. The file must use UTF-8 encoding. For validation details, see the <i>Infor ION Development Guide</i> .
+---- Object1.properties.json	FILE: A JSON file that describes additional properties for this object. It has the name of the object and a ".properties.json" extension. The file must use UTF-8 encoding and must be a valid JSON file according to the schema for the properties file. For details of this schema, see the <i>Infor ION Development Guide</i> .
+-- ...	Other objects of type DSV. Only one DSV schema file for each object must be used.
CustomHeaders	FOLDER: This folder contains the custom headers JSON file.
+-- CustomHeaders.json	FILE: A JSON file that includes the custom message headers to be added to objects that are stored in the Data Catalog.  For details on this schema, see the <i>Infor ION Development Guide</i> .

For details about creating metadata for custom objects, see the *Infor ION Development Guide*.

## Exporting objects

- 1 Select **Data Catalog > Object Schemas**.

2 Select one of these views to select objects.

- In tiles view you can use these select modes:
  - Single-select mode: Move the pointer over the tile to see the action icons.
  - Multi-select mode: Select one or more objects from the list
- In grid view, select one or more objects from the list.

3 Click **Export** to export the objects to a local file archive.

The archive file is downloaded to the download location of the browser. If only one object schema is selected for export, the archive file matches the name of the object. If multiple objects are selected for export, the downloaded archive file is called `ObjectSchemas.zip`. The archive contains the definition files for all selected objects as described earlier.

**Note:** You cannot export standard BODs.

## Deleting objects

1 Select **Data Catalog > Object Schemas**.

2 Select one of these views to select objects:

- In tiles view you can use these select modes:
  - Single-select mode: Move the pointer over the tile to see the action icons.
  - Multi-select mode: Select one or more objects from the list
- In grid view, select one or more objects from the list.

3 Click **Delete** to remove these objects from the Data Catalog.

A confirmation dialog is displayed.

4 Click **Yes** to confirm the deletion.

The selected objects are removed from the Data Catalog and the list is refreshed. You cannot undo a Delete action.

**Note:** You cannot delete standard BODs or AnyDocument.





## Viewing Object Metadata

In the Data Catalog, you can view the details of an object schema by performing one of these actions:

- In tiles view, click the tile of the object to view the details for.
- In grid view, click the drill-down icon located in the Details column.

This page is read only.

This table shows the areas of the object details page:

Area	Description
Header Tabs	Tabs used to display different information about the object. The details on each tab are explained in separate sections.
Toolbar	Contains the action icons available for the object.
	 <b>Back</b> – returns user to the Object Schemas page.
	 <b>Save</b> – saves changes that have been made to the object. <b>Note:</b> Only items on the <b>Configuration</b> tab can be edited at this time.
	 <b>Refresh</b> – refreshes the current page.
	 <b>Export</b> – exports the object metadata
Summary Pane	Contains the summary of the object, including the notifications pane for errors and warnings. The summary pane is located on the left side of the canvas. You can collapse this pane.
Main Canvas	Shows the relevant information for the object based on the selected header tab.

### Notifications pane

The notifications pane shows errors and warnings that have been found when making changes to an object. Summary counts are provided to show the total number of errors and the total number of warnings found. Details are also provided for each error and warning to assist in resolving the issues. All errors must be resolved before a **Save** action can take place.

For example, an error is reported if a custom header is added with an invalid name.

## Schema tab

The **Schema** tab shows the object schema. This tab has two secondary tabs: **Formatted View** and **Raw View**.

### Formatted View

The formatted view shows a list of schema properties and elements defined for the object.

A data grid is used to display important details for each property. The data that is shown in the grid is derived from the keyword values provided in the schema.

Paging is available for flat JSON and DSV objects. Use the paging control below the data grid to navigate through the complete list of properties.

### Property





The name of the property. For BODs, it is the name of the element or attribute.

**Indicators**

This column is displayed if the schema is registered with additional metadata properties. This column contains an identifier path, variation path, timestamp path, or delete indicator. Icons are used to represent each of the properties in the file.

This column does not apply to BOD objects.

This table describes the different indicators:

Indicator Name	Icon	Description
Identifier Path		Indicates that the property is used as an identifier for the object. Multiple properties can be defined to make up the unique identifier.
Variation Path		Indicates that the property is used to determine the variation for the object.
Timestamp Path		Indicates that the property is used to determine the last update or create time for the object.
Delete Indicator		Indicates that the property is used as the delete indicator for the object.

For more information on the additional metadata properties file, see "Defining additional object metadata properties" in the *Infor ION Development Guide*.

**Title**

Shows the title for each property, if defined in the schema. If none of the properties in the schema contain a title, this column is hidden by default.

This column does not apply to BOD objects.

**Description**

Describes the property in the schema. If none of the properties in the schema contain a description, this column is hidden by default.

**Data type**

The data type of the property.

**Position**

This column shows the order in which the properties in the data object should be displayed to the user. By default, the properties on this page are listed in ascending order according to this value. Any properties that do not have a position defined in the schema are listed after those that do. If none of the properties in the schema contain this value, the column is hidden from the page.

This column does not apply to BOD objects.

**Required**

Indicates that the property must be included in the payload. This column is hidden for DSV objects.

**Repeatable**

Visible for BOD objects only. Indicates that the element can occur more than once in the data object.

**Length**

This column is used for string properties only. It specifies the maximum length that the value in the payload can be. If there is no maximum length defined for the property, "Not Provided" is shown in the cell.

**Digits Before Decimal**

This column is used for number properties only. It specifies the maximum number of digits that is allowed before the decimal for the payload value. If there is no maximum value provided for the property, “Not Provided” is displayed in the cell.

**Digits After Decimal**

This column is used for number properties only. It specifies the maximum number of digits that are allowed after the decimal for the payload value. If this is not defined for the property, “Not Provided” is displayed in the cell.

**Negatives Allowed**

This column is used for number properties only. A check box is used to indicate whether negative values are allowed for the property in the payload. If the minimum value that is provided in the schema includes negative numbers, or the minimum is not provided, the check box is selected.

**Date Time Format**

This column is used for date, time, and datetime properties only. For properties that use a standard format, “Standard” is displayed in this column. When a custom format is used, the exact format as defined in the schema is displayed. For BODs, all date, time, and datetime properties are considered “Standard.”

**Note:** For JSON and DSV objects, if you pause on “Standard”, a tooltip with the expected format is displayed.

To view a property as defined in the schema, a right panel has been added to the page. This panel is collapsed by default. To expand it, click the icon that is located at the top of the panel. After you have selected a row, the panel shows the selected row's property definition in its raw format. This panel is not yet supported for BOD objects.

**Raw View**

The raw view shows the full object schema in its raw format. To search the schema on this tab, click anywhere inside the main area of the **Raw View** tab and press **Ctrl + F**. A search bar, where you can specify the text you want to find in the schema, is displayed.

For BOD schemas that include other XSD files, the **Raw View** tab is a drop-down menu. A list of all included schema files, and the main file itself, are displayed in the drop-down. You can also click the name of an included schema as an alternative way to navigate to a different schema's raw view.

## Personalize Columns

By default, all columns are displayed in the data grid.

If you do not want to see all the columns, click **More...**, the ... above the data grid and select **Personalize Columns**. A dialog box is displayed with the list of columns. Clear the check box of the column that you want to hide from the page. Click **OK** to close the dialog box.

## Filter Schema Properties

Column level filtering is available with the data grid. If the filter row is hidden, click the **More...** icon and select **Show Filter Row**. When the filter row is shown, you can filter on any of the columns to find the information you are searching for. To clear all filters, click **More...** and select **Clear Filter**. To hide the filter row, click **More...** and select **Show Filter Row**.

**Note:** There may be a property that is defined as both the “Identifier Path” and the “Variation Path”. When filtering on the **Indicators** column, you must select the Identifier Path + Variation Path option to see that property when running the filter.

## Dialect tab

The **Dialect** tab is only visible for DSV objects. This tab describes the format of the delimited file.

These fields are shown:

### Separator

The character used to separate the values in a row of data.

### Header Rows

The number of header rows included in your payload before reaching the first row of data.

### Column Header Line

The row number that contains the column headers for the data object.

### Enclosing Character

The character used to indicate the start and end of a value.

## Noun properties

The **Noun Properties** tab is only visible for BOD objects.

This tab shows the contents found in the noun properties XML file associated with the BOD.

### Identifier Path

The XPath to the element that contains the unique identifier for the BOD object.

### Accounting Entity Path

The XPath to the element that contains the accounting entity associated with the BOD instance.

### Location Path

The XPath to the element that contains the location associated with the BOD instance.

### Status Path

The XPath to the element that contains the status code for the BOD instance.

### Relations

Shows the list of relationships between the current BOD noun and other nouns.



This table shows the column name with the description:

Column name	Describe
To Noun	The noun that the current BOD noun is related to.
From Path	XPath to the element of the current BOD noun that is used to map it the related noun.
To Path	XPath to the element of the related noun that is used to map it to the current BOD noun.

## Object Properties

The **Object Properties** tab is only visible for JSON and DSV objects.

This tab shows contents found inside the additional metadata properties file called `<object name>.properties.json`. If there are no additional metadata properties defined for an object, this tab is hidden.

### Identifier Paths

The JSON paths to the properties that uniquely identify the object.

### Variation Path

The JSON path to the property that contains the variation value for the object.

### Delete Indicator

The JSON path to the property that is used to indicate whether the object has been marked as deleted.

### Delete Indicator Value

The value that the delete indicator must have to indicate that the object has been marked as deleted.

### Timestamp Path

The JSON path to the property that contains the timestamp of the moment the object was created or updated.

### Additional Properties

Shows the information found in the `AdditionalProperties` node in a tree grid. The **Name** column shows the name of the element. The **Value** column shows the value for the element.

For more information on the additional metadata properties file, see "Defining additional object metadata properties" in the *Infor ION Development Guide*.

## Configuring custom message headers

Custom message headers for objects that are processed by ION Services are managed in the Data Catalog.

For information on message headers, see "Message Headers" in the *Infor ION Development Guide*.

When defining custom headers, the header names must follow these conventions:

- The name must start with the `Custom_` prefix.
- The name cannot exceed 250 characters.
- Only alphanumeric characters are allowed. The only special character that can be used is the underscore that separates the `Custom` prefix from the user-defined header name.
- Spaces are not allowed.
- Names must be unique within each object.
- The name is case-sensitive. For example, `Custom_BankID` and `Custom_bankid` are two different headers.

## Importing custom headers

To import custom headers into the Data Catalog, these elements must be included in the zip file that is used to import object metadata:

- A `CustomHeaders` folder
- A `CustomHeaders.json` file

For information on the folder structure, see [Zip file structure for import](#) on page 466.

- 1 Select **Data Catalog > Object Schemas**.
- 2 Click **Import** to upload custom headers.
- 3 Browse to the local file archive that contains the custom header definitions. Click **OK** to upload this file.

After the upload is complete, a message is displayed, showing the result of the import. If there are any errors, you must correct them and try to import again.

If the import is successful, any existing custom headers for the specified objects are overwritten by the content that is provided in the file.

**Note:** Any objects that are included in the custom headers file that do not exist in the Data Catalog are skipped during the upload.

## Exporting custom headers

- 1 Select **Data Catalog > Object Schemas**.
- 2 Select one or more objects to export.  
In the tiles view, move the pointer over the tile to export a single object. Alternatively, enable the multi-select mode to select more than one object.

- 3 Click **Export** to export objects and their custom headers to a local file archive.

The archive file is downloaded to the download location of the browser. The file includes the metadata and custom headers of the selected objects.

**Note:** Although standard BOD metadata cannot be exported, custom headers for selected standard BODs are included in the exported `CustomHeaders.json` file.

## Adding custom headers

**Note:** You can define a maximum of three custom headers per object. BODs are grouped by noun in the Data Catalog. Therefore, you can define up to three custom headers for each BOD object within a noun.

- 1 Select **Data Catalog > Object Schemas**.
- 2 Select an object from the page and click to view its details.
- 3 Click the **Configuration** tab.
- 4 Click **Add** to add a custom header for the object.
- 5 After the new row is added, complete these steps:
  - a BOD only: In the **Verb** column, select a verb from the drop-down menu to determine to which object the new custom header is added.  
For example, if you add a custom header to `Sync.SalesOrder`, you must select **Sync** from the drop-down on the SalesOrder BOD details page.
  - b Click inside the **Name** column to specify a name for the custom header.
  - c In the **Data Type** column, select the expected data type for the value of the custom header.
- 6 Click **Save**.

## Updating custom headers

- 1 Select **Data Catalog > Object Schemas**.
- 2 Select an object on the page and click to view its details.
- 3 Click the **Configuration** tab.
- 4 In the custom headers data grid, click inside the cell of the value you want to change.
- 5 After the change has been made, click **Save**.

## Deleting custom headers

- 1 Select **Data Catalog > Object Schemas**.
- 2 Select an object on the page and click to view its details.
- 3 Click the **Configuration** tab.
- 4 Select one or more custom headers to delete.
- 5 Click **Delete**.
- 6 Click **Save**.

## User area extensions

Navigate to **Data Catalog > Schema Extensions**. On this page you can upload schema files to be used as user area extensions in the context of this tenant. You can also define the mappings between

the standard nouns and the extensions to use. For details about how to define user area extensions, see "Using a custom XML structure in the UserArea" in the *Infor ION Development Guide*.

The custom metadata imported in these screens is stored in the Data Catalog and is available for all users of this tenant.

This page contains two tabs:

- **Extensions** - to specify which nouns will use the user area extensions
- **Schema Files** - to manage metadata for user area extension definitions

See the following sections for a detailed explanation of all operations available in these screens.

## Importing user area schema files

- 1 Select **Data Catalog > Schema Extensions** and click the **Schema Files** tab.

This information is displayed:

### **Used**

Flag indicating whether the schema file is already mapped to a User Area.

### **File Name**

The name of the schema file. File names are unique within a tenant.

### **Last Updated by**

The UPN identifier of the user who performed the last upload for this noun.

### **Last Updated on**

The date and time this noun was uploaded.

To refresh the list of custom nouns, click this button:

 - **Refresh**

- 2 Click **Import** to upload a new schema file.
- 3 Browse to the schema definition file (XSD) on the local drive and select the file to import. Click **OK** to upload this file to the Data Catalog or click **Cancel** to abort the import operation.

After a while a message box is displayed, showing the result of the import. If there are validation errors, you must correct the definitions and try to import again. If all the validation rules passed, the schema files are imported and added to the list. The list of schema files is automatically refreshed after an import.

If a schema file name already exists, a message box is displayed, asking you to confirm that the uploaded file overwrites the previous definition. The file name comparison is case-insensitive.

**Note:** If you update a schema file that is in use, the mappings for the standard nouns using this schema do not need to be updated. However, if there are active monitors, alarm templates, activation policies, or object flows that use the elements from this user area extension, these must be re-activated. Ensure to inform all tenant users when a schema is updated so that they can re-activate their configurations.

---

## Exporting user area schema files

- 1 Select **Data Catalog > Schema Extensions** and click the **Schema Files** tab.
- 2 Select the schema file to export and click **Export**.

The schema file is downloaded to the download location of the browser.

## Deleting user area schema files

- 1 Select **Data Catalog > Schema Extensions** and click the **Schema Files** tab.
- 2 Select the schema file to be deleted. You can only delete schemas that are not used in the extensions mappings; these are marked with a warning icon.
- 3 Click **Delete** to remove the selected schema from the Data Catalog. A confirmation dialog box is displayed.
- 4 Click **Yes** to confirm the deletion.

The selected schema is removed from the Data Catalog and the list is refreshed. You cannot undo a Delete action.

## Mapping user area extensions

To extend standard nouns with user area extensions:

- 1 Select **Data Catalog > Schema Extensions** and click the **Extensions** tab.

This information is displayed:

### **Noun**

The name of the standard noun to which a user area extension is mapped.

### **XML Path**

The XPath expression of the UserArea.

### **Schema**

The name of the schema file that is attached to it.

To refresh the list of custom nouns, click this button:

 - **Refresh**

- 2 Click **Add** to specify user area extensions mappings for a new noun. The **Add User Area Extension** dialog box is displayed.
- 3 In the dialog box, select the noun you want to extend. All standard nouns available in all libraries in the Data Catalog are displayed. You cannot extend a custom noun; therefore the custom nouns are not displayed in this list.
- 4 Once a noun is selected, all the XML Paths where an extended UserArea element could be attached are listed. Select the path to the element you want to extend.

- 5 Select the schema to attach to the path you selected previously. All schema file names uploaded for this tenant are displayed and can be selected. You can only attach one schema file to a selected path.
- 6 Click **OK** to confirm this configuration. The new mapping is added to the Data Catalog and the list of extensions is automatically refreshed. The user identifier and the timestamp of the last update are displayed in the upper-right corner of the screen.
- 7 To add more mappings, repeat the steps above.

## Removing user area extensions mappings

- 1 Select **Data Catalog > Schema Extensions** and click the **Extensions** tab.
- 2 Select the mapping you want to delete. Click **Remove**. A confirmation dialog box is displayed.
- 3 Click **Yes** to confirm the deletion. The mapping is removed from the Data Catalog and the list is automatically updated. The schema file used by this mapping is not deleted.

## Editing user area extensions mappings

- 1 Select **Data Catalog > Schema Extensions** and click the **Extensions** tab.
- 2 Select the mapping you want to change and click **Edit**.
- 3 In the editing dialog box that opens, you can only change the name of the schema file to use. Select a new schema file and click **OK** to confirm the change. The change is automatically uploaded to the Data Catalog and is available to all users of this tenant.

**Note:** If there are active monitors, alarm templates, activation policies, or object flows that use elements from the changed user area extension, these must be deactivated and changed to use the new user area extension definition.

## Locale selections

You can use the **Locale Selections** page to configure the locales that the applications in your CloudeSuite support. This table is cross-referenced by the JDBC driver when querying the Data Lake for localized data. If your application is not localized, the locale selections that are configured on this page are not used.

For information on Data Lake queries for localized data, see the *Infor ION Development Guide*.

To access the **Locale Selections** page, you require one of these ION Desk permissions:

- View  
You can only view and export the locale selections.
- All

---

You can perform all operations available on the page.

On this page, this information is displayed:

### Position

The locale position, as it relates to the numbered columns in your data store used for localized data.

### Locale Code

The locale code supported by your application. The locale codes added to this page must match the locales used in the data objects containing localized values.

Examples:

- en\_US (two-character language code with the two-character country code).
- en ( two-character language code)

### Locale Code Search List

A comma delimited list of alternative locale codes to use when the locale code that is selected is not available in the source application. The search list is stepped through, in order, until a locale code match is found. If the entire list is exhausted without finding a match, the locale code that is located in Position 1 is used by default.

Example: en\_US, en, en\_GB

## Adding locale selections

- 1 Select **Data Catalog > Locale Selections**.
- 2 Click **Add (+)** to add a new row to the table.
- 3 Click inside the **Locale Code** column and add your new locale code.
- 4 Click inside the **Locale Code Search List** column. Add the alternative locale codes to be used when the main locale code is not available:  
Ensure that the list starts with the locale code you specified in the **Locale Code** column.
- 5 Click **Save**.  
A warning message is displayed informing you that changing your locale selections has significant effect on your localized data.
- 6 Click **Save** to save your changes.  
**Note:** You can add 32 locales at most.

## Editing locale selections

- 1 Select **Data Catalog > Locale Selections**.
- 2 Select the locale selection to edit.  
Click inside either the **Locale Code** column or the **Locale Code Search List** column. The field becomes editable.

- 3 Specify a different locale code or list of locale codes.
- 4 Click **Save**.  
A warning message is displayed informing you that changing your locale selections has significant impact on your localized data.
- 5 Click **Save** to save your changes.

## Deleting locale selections

- 1 Select **Data Catalog > Locale Selections**.
- 2 Search for the locale selection to delete and select the check box in that row.
- 3 Click **Delete** to remove the selected locale from the list.
- 4 Click **Save**.  
A warning message is displayed informing you that changing your locale selections has significant impact on your localized data.
- 5 Click **Save** to save your changes.

## Repositioning locale selections

- 1 Select **Data Catalog > Locale Selections**.
- 2 Click and hold the handle icon of the locale selection to reposition.  
You can find the icon in the first column.
- 3 Drag and drop that row to a new position in the list.
- 4 Click **Save**.  
A warning message is displayed informing you that changing your locale selections has significant impact on your localized data.
- 5 Click **Save** to save your changes.

## Importing locale selections

- 1 Select **Data Catalog > Locale Selections**.
- 2 Click **Import** to import a list of locale selections.  
**Note:** Importing a list of locale selections overwrites what currently exists on the locale selections page.
- 3 Browse to the local file archive and select the file to import.  
This file must be a JSON file.
- 4 Click **OK**.



When the file is properly imported, the list of locales is displayed.

**5** Click **Save**.

A warning message is displayed informing you that changing your locale selections has significant impact on your localized data.

**6** Click **Save** to save your changes.

## Exporting local selections

**1** Select **Data Catalog > Locale Selections**.

**2** Click **Export** to export a list of locale selections.

The locale selection file is saved on the download location of the browser.

**Note:** When exporting locale selections, the entire list is exported. You cannot select a subset of locales from the list.

For information on how to define metadata for localized strings, see the *Infor ION Development Guide*.

---

## Chapter 16: Data Lake

Infor Data Lake provides raw storage to capture transactional and informative events.

Any type of content can be sent and stored in Data Lake, including this content:

- BODs (Business Object Documents).
- JSON.
- Delimited file formats such as CSV, TSV, and PSV.
- Binary file formats using the Data Catalog's `Any` data format.

To ensure data security, all data objects that are stored in Data Lake are encrypted with AES-256 bit encryption. Data in-transit is secured by TLS version 1.1.

Content that is captured by ION provides enriched services to help identify Data Lake receipt and troubleshoot issues. This can be helpful when providing large volume data publishing for objects stored in OneView before availability in Data Lake.

See [Data Lake flows](#) on page 36.

Objects can be retrieved from the Data Lake by Birst for analytics strategy and development or with API methods exposed in the ION API Gateway.

See *Infor ION Development Guide*.

Additional tools and APIs are available to manage the data that resides in Data Lake. You can demote objects to long-term storage by ad hoc or policy-based management schemes in these circumstances:

- To help optimize query loads.
- Reduce storage utilization.
- To maximize available frequent access storage

Data can be restored to the frequent access layer of Data Lake with the restore interfaces.

The Data Lake menu includes these sub-menus:

- **Data Lake overview**  
Displays the storage usage statistics of Data Lake.
- **Storage policies**  
Defines and manages storage policies and data object subscriptions. Updates to policies are captured in the policy logs.
- **Restore**  
Restores and promotes filter-based data objects from archive to the Data Lake's frequent access layer. Persistence criteria are captured in the logs.

- **Purge**  
Removes data objects from the Data Lake based on filtering criteria or specific data object ID(s).

## Data Lake overview

The **Data Lake overview** menu displays the storage statistics of Data Lake. Widgets display the storage distribution by data object type. The top data objects are displayed by name and count of ingested data.

Name	Description	Filters
Overview	Shows the total number and size of data objects stored in Data Lake.  Overview of the data object distribution in Data Lake by type.	-
Top objects	Shows the top data objects by name and size.	Filter by Time: <ul style="list-style-type: none"> <li>• All Times.</li> <li>• Last 24 hours.</li> <li>• Last 7 days.</li> </ul> Filter by data object type.
Data ingestion	Count of messages that are captured in Data Lake during a certain period.	Filter by Time: <ul style="list-style-type: none"> <li>• All Times.</li> <li>• Last 24 hours.</li> <li>• Last 7 days.</li> </ul> `

## Storage policies

Data in the Data Lake is stored in a frequent access layer ensuring quick access. Storage policies can be applied to data residing in the Data Lake that becomes less frequently used or rarely accessed. This includes BODs, JSON, and `Any` Data. After a policy is enacted, data is demoted to a long-term storage class.

When eligible, any data objects ingested to Data Lake have a storage policy applied. Accordingly, objects are automatically demoted to long-term storage unless configured otherwise.

Often, objects or documents that are captured in Data Lake are used in integration solutions and modeled within ION's data flows with Data Lake flow. Documents, messages, or objects that are

modeled in the Data Lake flows are subject to the storage policy. If a document was exposed to both Document flow and Data Lake flow it is automatically subject to the storage policies.

Any changes to any storage policy affect new content that is captured to Data Lake. Any content already residing in the Data Lake preserves its previous policy settings.

The storage policy demotes the data to the long-term storage class. This class persists data that is rarely accessed but requires persistence for archival and historical purposes.

When data is demoted from the frequent access layer to the Long-term storage layer. The data is subjected to a 30-day transition period when the data is archived. Access to data is restricted until this transition period has elapsed. You can restore the data with the **Restore** menu.

## Archive criteria

You can define the storage policy per single or multiple data object(s). Specify the relative age of data to archive in terms of days. Single data objects reside in one Storage Policy. All entities that are older than the specified time are archived to the long-term storage class.

After the data is stored the storage class contains these age criteria:

- OFF
- 30
- 60
- 90
- 180
- 270
- 365
- 730

By default, there is a single Storage Policy defined. This table shows the specified criteria for the Default Storage Policy:

Name	Default policy
Description	All data objects that have no policy defined.
Age selection	OFF
Data objects	All

Before storing data in Data Lake, configure the Default Storage Policy or define new custom storage policy. By default, the archive age of data objects in the Default Storage Policy is set to OFF.

**Note:** Age of data regards the date when data was stored into Data Lake, not the date from the data content. After new changes are applied to the Storage Policy, it applies only to new stored data.

## Audit logs

When adding new storage policy or update an existing policy, an event impression is captured in the logs. To access the Audit Logs, go to **Data Lake > Storage Policy > Audit Logs**.

Audit logs display this information:

- Owner of the update.
- Date of update.
- Storage policy name.
- Event.

This table lists the events:

Name	Description
Policy was created	When new policy is created
Data Object “xyz” was added from the Policy	When the data object was attach to new storage policy.
Data Object “xyz” was removed from the Policy	When data object was removed form storage policy
Age for Storage Classes was change in the Policy	When age for demoting the data object into Long-term storage was changed.
Policy was deleted	When storage policy was deleted

## Setting up storage policies

- 1 Click the hamburger icon to open the menu.
- 2 Select **Data Lake > Storage Policies**.
- 3 Click **Add** from the toolbar.  
The details page of the storage policy is displayed.
- 4 Click the **Settings** tab.
- 5 Specify this information:
  - Name
  - Description
  - Age of Data for long-term storage
- 6 Click the **Data Object** tab.
- 7 Click **Add** to add data objects.
- 8 Click **Save**.

**Note:** The selected data object is now subject to the new storage policy. This change applies to new created data for specified data object name.

You can define multiple Storage Policies at once.

---

## Updating a storage policy

- 1 Click the hamburger icon to open the menu.
- 2 Select **Data Lake > Storage Policies**.
- 3 Select single storage policy from the list panel on the left side.  
The details of selected storage policy are displayed.
- 4 Update this information:
  - Name
  - Description
  - Age
- 5 Select new data objects.
- 6 Click **Save**.

**Note:** Updates to the storage policies are applied to any new data objects that are captured in the Data Lake. Objects that were captured before policy updates maintain their policy at time of capture.

## Deleting a storage policy

- 1 Click the hamburger icon to open the menu.
- 2 Select **Data Lake > Storage Policies**.
- 3 Select the storage policy form the list panel.
- 4 Click **Delete** in the toolbar.

**Note:** you cannot delete the default policy. Only updates to policy settings are managed or updated. Data objects that were attached to the deleted storage policy are now subject to the default policy.

## Restoring data objects

After data objects are demoted to long-term storage they are not accessible to any data retrieval interface. This includes the Data Lake's Payload API methods, Compass API methods, Birst, and other applications. You must restore the data object from archive.

To restore data objects, you must specify these search criteria:

- **Date Range From** and **Date Range To** - Searches for objects that were stored in Data Lake in that time range.
- **Data Object(s)** names - Select the names from the drop-down list to restore from long-term storage.
- The restore period for how long the data objects must be kept in the frequent access storage:
  - For number of days - Restores the data objects for the specified number of days. After those days have passed, the data objects are moved back to Long Term Storage
  - Permanently - The data objects are restored permanently.

The restore logs panel shows the details of the restore request and the status of the restored data objects. You can use the panel splitter to increase or decrease the height of the panel.

The logs details show these details:

- **Restore Filter** - The data objects filter that was used to start the restore request.
- **Status** - The status of the restore procedure.
- **Available for** – For how many days the data objects are restored.
- **Date Range From / Date Range To** – The time range filter used for the restore request.
- **Data Object Count** - Number of data objects that were restored.
- **Request Date** – Time when the restore request was submitted.
- **Requested by** – User that submitted the restore request.

### Restore procedure status

A restore procedure can have one of these statuses:

- **PENDING** - The restore procedure is initiated and the data objects start getting promoted to the frequent access storage. This procedure takes more than 6 hours.
- **AVAILABLE** - The data objects were promoted to the frequent access storage and are available until they get demoted back to the long-term storage.
- **EXPIRED** - The data objects were demoted back to the archive.

### Limitation in current version

Permanently restored data objects cannot be moved back to the long-term storage.

You cannot restore more than 500 data objects in a restore request. We recommend that you refine the date range and data objects selection to request for a smaller set of data objects to be restored. Or you can use the Data Lake Restore API in ION API.

See the *Infor ION Development Guide*.

## Purge data

On the purge page you can remove unwanted data objects that are stored in both frequent access storage and long-term storage from Data Lake.

Purged objects cannot be retrieved from Data Lake. Purging of data objects can be done in these ways:

- Use advanced filters to search for data objects to purge.
- Use the unique data object ID(s).

After a purge event starts it takes some time to complete.

You cannot revert an active purge event. Nonetheless, a purging process can be stopped by a user to prevent any further data objects from being purged. Any objects that were already purged before stopping the process cannot be restored and are permanently removed from Data Lake. If the event

is not stopped, the purge activity continues to run until all objects defined by the purging parameters are removed from Data Lake.

Stopping a purge event results in a partially completed purge. You can verify the status of a purge event in the purge logs.

**Note:** Purging data objects from Data Lake deletes the data object payload from ION OneView.

## Purge by ID

Depending on the number of objects to be purged, this activity can take an extended period to complete. Current active or completed purge events are displayed in the Purge logs.

See [Purge logs](#) on page 489.

- 1 Click the **Purge by ID** option.
- 2 Specify the required **Data Object ID** field with the ID(s) of the objects to purge.  
Multiple Object ID(s) can be used for filtering, separated by a comma.
- 3 Click **Purge**.  
A confirmation message is displayed, with the total size and total number of objects to be purged. Optionally, you can add a description to be displayed in the logs section.
- 4 Click **YES** to confirm the Purge action.

## Purge by filter

Depending on the number of objects to be purged, this activity can take an extended period to complete. Current active or completed purge events are displayed in the Purge logs.

See [Purge logs](#) on page 489.

- 1 Click the **Purge by Filter** option.
- 2 Specify these filters:  
**Date Range from and Date Range to**  
The date and time range filter searches for data objects that were stored in Data Lake during the selected period.  
**Data Objects**  
Filter by data object name(s), as they are registered in the Data Catalog to purge. For example, `Sync.SalesOrder`.
- 3 Optionally, specify these filters:  
**Accounting Entity**  
Accounting entity of the business object in the data object.



**Logical IDs**

Identifier of the provider of the data object.

**Location**

The location of the business object that is in the data object.

**4 Click Purge.**

A confirmation message is displayed, showing the size and number of the data objects to be purged. Optionally, you can add a description before confirming.

If some of the filters are invalid the fields are indicated with an error message.

**5 Click YES to confirm the Purge action.**

## Purge logs

You can view every purge event that is complete or in-progress in the Purge Logs section on the **Purge** page.

The current running purge events are displayed in the top row(s) of the log pane. The rest of the logs are sorted in chronological order of when they were started.

The purge logs contain both, Purge events that were started from the **Purge** page in ION and the Data Lake Purge API.

This information is displayed:





- **Purge Status**  
Displays the current progress of the purge action. Purge Status section covers the status indicators.
- **Criteria**  
The filter criteria that was used to specify the data objects and apply purge action on them.
- **Total to Purge**  
The total number of objects that matched the filter criteria that was used to start the purge action.
- **Total Size in Bytes**  
The total size in bytes of the data objects that were purged.
- **Objects Purged**  
The total number of objects that were removed by the purge action.
- **Start Date**  
The date and time when the purge event was started.
- **End Date**  
The date and time when the purge event was completed.
- **Source**  
Indicates from where the purge event was initiated. The purge event can be started from ION Desk or through the Infor Data Lake API.

- **Purge Description**  
The user defined description that was specified when confirming the purge action.
- **Purged By**  
The user who started the purge action.

You can drag the horizontal splitter to increase or decrease the height of the logs panel.

## Purge Status Indicators

This table shows the status indicators that specify the current progress of a purge event:

Icon	Status	Description
	In Progress	Indicates a purge event in progress. Purge in progress can be stopped.
	Partially Completed	Indicates a purge event that was stopped while in progress. Can be partially completed.
	Completed	Indicates a purge event that completed successfully without being stopped.
	Error	Indicates a purge event that contains errors and was not completed.

Click the error status row to display the error message.

## Stopping a Purge event

Whenever a purge process starts it is shown at the top of the **Purge Logs**. A purge in progress icon is shown in the status column.

Stopping a purge does not mean it is canceled and that the removal action can be undone. All the already removed data objects cannot be retrieved back from Data Lake

To stop the purge in progress:

- 1 Select the row that indicates there is a purge event in progress.
- 2 Click **Stop the Purge**.
- 3 Confirm to stop the purge process.  
The status of the purge changes to partially completed. It shows the total number of data objects that were removed, until the purge event was manually stopped.

---

## Compass UI

Compass is an ION Desk tool for querying the contents of Data Lake for data analysis & exploration, ad-hoc data exports, data management, and troubleshooting.

**Note:** Data Lake Compass page is not available in AWS GovCloud.

In Compass, you can run these tasks:

- Write and run SQL queries.
- Export data.
- Explore data objects and properties.
- Execute administrative tasks within Data Lake.

## Overview

Compass offers these elements and functionality:

- Multi-tabbed query development with up to 10 concurrent tabs.
- Data Objects Panel - for exploring Data Catalog objects.
  - Ability to search and filter data objects and data object properties.
  - Generate contextual SQL statements and SQL hints to influence data retrieval behaviors.
  - Generate contextual administrative SQL code for managing & troubleshooting data residing in Data Lake.
- Query Editor - for writing SELECT statements and running them.
  - Syntax highlighting.
  - Contextual auto-suggest for MS SQL keywords, custom keywords, data object and data object property names.
  - SQL formatting
  - Font size controls.
- Results Panel - for previewing the query results.
  - Success and error messages.
  - Result set in a data grid.
  - Exporting a result set.

## Open Compass

- 1 Select ION Desk.
- 2 Expand the Data Lake section
- 3 Click **Compass**.

---

## Data objects panel

With the data objects panel you can browse through the data object definitions in Data Catalog that are queryable within Data Lake.

The Compass supported data object types:

- Newline-delimited JSON (NDJSON)
- Delimiter-separated values (DSV)
  - Comma-separated values (CSV)
  - Tab-separated values (TSV)
  - Pipe-separated values (PSV)
  - User-defined DSV

## Data object tree

The data object tree lists the compatible data object schemas that are registered in Data Catalog.

Expanding a data object shows the types and names of the object properties. The icons in front of object and properties indicate these types:

- Object type
- Property type

When hovering over data objects and object properties, an ellipsis icon is displayed. Clicking the ellipsis icon presents contextual options for the selected data object or properties and allow you to accelerate code generation and syntax development.

## Refreshing the list of Data Objects

Each time Compass is loaded for the first time; the list of data objects is retrieved from Data Catalog. Each time you expand a data object, by clicking the + icon, the list of properties for that object are retrieved from Data Catalog.

Expanding a data object to get the list of properties also help with preloading the property names in the SQL auto-suggest keywords list.

When expanding an object within the data object navigator, properties are cached during a user's active session within Infor OS. In the event that updates to the object's metadata are made, refreshing the tree reloads the metadata definitions from the Data Catalog.

To refresh all data objects, click the refresh icon above the data object tree.

To refresh one data object and its properties:

- 1 Hover over the data object.
- 2 Click the ellipsis menu.
- 3 Click **Refresh Object**.

## Filtering the tree

To find a data object in the tree, you can search by keyword or filter by object type.

### Searching by keyword

- 1 Expand the left data object panel.
- 2 Click the search box.
- 3 Specify the text.  
Objects matching the criteria are shown and the keyword is highlighted.

### Filtering by object type

- 1 Click the filter icon
- 2 Specify the the data object type.  
Data objects matching the filter criteria are shown.

## Data object menu

Hovering over a data object or property displays an ellipsis icon. Click the ellipsis icon to show the Data Object Menu. These options are shown:

- Expand – Expands the data object and shows the object's properties.  
This option is not displayed in the menu for the object properties.
- Refresh – Retrieve the latest schema definition for the data object.
- Generate SELECT – Inserts a generated SELECT SQL statement in the code editor.
- Generate Admin – Inserts a Compass administrative stored procedure for the object name.  
This option is not displayed in the menu for an object property.
- Open in Data Catalog – Opens the Data Catalog object page.

## Generating SELECT

Generated SELECT has different variants depending on what data to retrieve. The default SELECT option must be used in most cases, as it retrieves the latest data.

In cases where the latest variation of the data do not suffice, use the Generate SELECT with Hints (--\* statements).

---

For more details on Hints functionality, see the "Data Lake Queries" section in the *Infor ION Development Guide*.

- 1 Pause on a data object or data object property.
- 2 Click the ellipsis menu.
- 3 Select **Generate > SQL**
- 4 Select one of the available SELECT options.

Clicking one of the options in the ellipsis menu, generates and inserts a SELECT statement in the query editor. The generated SELECT statement includes all the object property names, and the object name.

## SELECT options

### SELECT

```
SELECT "Order_TSV"."Customer", "Order_TSV"."Order", "Order_TSV"."OrderDate  
time", "Order_TSV"."OrderNumber" FROM "Order_TSV"
```

### SELECT (All Variations)

```
--*includeDeletionsWithMaxVariations=Order_TSV  
SELECT "Order_TSV"."Customer", "Order_TSV"."Order", "Order_TSV"."OrderDate  
time", "Order_TSV"."OrderNumber" FROM "Order_TSV"
```

### SELECT (Include Deleted Rows)

```
--*includeAllVariations=Order_TSV  
SELECT "Order_TSV"."Customer", "Or  
der_TSV"."Order", "Order_TSV"."OrderDatetime", "Order_TSV"."OrderNumber"  
FROM "Order_TSV"
```

### SELECT (Include Additional Properties)

```
--*includeInSelectAll=s/g/p/l  
SELECT * FROM "Order_TSV"
```

### SELECT (Skip Reformatting)

```
--*skipreformatting  
SELECT "Order_TSV"."Customer", "Order_TSV"."Order", "Order_TSV"."OrderDate  
time", "Order_TSV"."OrderNumber" FROM "Order_TSV"
```

---

## Generating administrative procedures

You can generate administrative procedures to manage the Compass data storage.

For more details on the administrative command functionality see the "Data Lake Queries" section in the ION Development Guide.

To generate an administrative procedure:

- 1 Hover over a data object name..
- 2 Click on the ellipsis menu.
- 3 Select **Generate > Admin**.
- 4 Select one of the available administrative commands:

### Clear Table – Remove Formatted Data

```
exec infor.clear_table('Order_TSV', 'true')
```

### Clear Table – Retain Formatted Data

```
exec infor.clear_table('Order_TSV', 'false')
```

### Clear Data

```
exec infor.clear_data('Order_TSV', 'YYYY-MM-DD')
```

### Clear View

```
exec infor.clear_view('Order_TSV')
```

### Reset Partitions

```
exec infor.reset_partitions('Order_TSV')
```

## Query editor

With Compass editor you can write simple or advanced SQL SELECT statements and run them to retrieve data from Data Lake.

These are the supported SQL statements:

- SELECT statements
  - MS SQL dialect.
  - SQL Comments are supported within a statement.
- SELECT statement with Hints, to retrieve data variations.
- Admin statements to manage the formatted data lake objects.

## Running a query

To run a query:

- 1 Click the code editor.

- 2 Write or paste the SQL statement.
- 3 To execute the SQL statement, click **Run Query**, or Ctrl + Enter.  
After clicking **Run Query**, the running Query ID is displayed. A timer starts counting the elapsed time until the results are retrieved and displayed in the results pane.  
The **Run Query** button changes to **Stop Query**. Click **Stop Query** to prevent the retrieval of the results.  
The results output is limited to maximum 100 rows of data. If you must retrieve larger result sets, use the Compass API.  
See the *Infor ION Development Guide*.

## Running multiple queries

To persist the results of an executed query to investigate more data by running a new query:

- 1 Open a new tab by clicking the **+** icon in the toolbar tab.  
You can have up to 10 open tabs at the same time, each having different SQL and results.
- 2 Click the **X** icon, to close a tab.

## Syntax and formatting

Compass uses MS SQL syntax highlighting and provides keyword suggestions as you type.

The keywords provided are based on MS SQL syntax, data object names, and properties from the Data Catalog. For the object property names to be suggested, you must expand a data object in the tree to fill the keyword list with its property names.

## Using the auto-suggest functionality

- 1 Start typing in the query editor.  
List of words is suggested.
- 2 Press Enter or click the word to auto-complete it.

## Formatting the query

With format you can apply formatting on the query text or change the formatting case of the SQL statement. Compass queries are case insensitive. Formatting is applied to have better readability of the queries.

To format the query:

- 1 Click **Format**.



2 Select one of these options:

**Format SQL**

Formats the SQL with new lines and indents.

**To Upper Case**

Changes the query text to upper case.

**To Lower Case**

Changes the query text to lower case.

3 To increase or decrease the editor text size, click the magnifying icon with the + or – symbol.

## Results panel

After you run a query, the results panel is shown at the bottom of the window.

These results are shown::

- Data results of successfully running a query.
- Error messages due to: invalid syntax, invalid object names or service respond errors.
- Output of running administrative procedures.

## Data results

When a query is successfully executed the results are displayed as a data grid.

The details for a successful query that are displayed in the panel indicate:

- Number of rows in the results. Maximum 100 rows of data is displayed.
- Elapsed time to retrieve and display the results.
- Date and time when the query was executed.
- Query ID for the results.
- Icon is displayed when new results are fetched if you executed a new query.

After the results are displayed, you have these options:

- Search the results.
- Export the results to CSV.
- Maximize the view.
- Change the row height of the data grid.

## Searching results

To search the data grid using a keyword:

1 Click the search box in the data grid toolbar.

- 2 Specify a keyword.
- 3 Press Enter to run the filter on the data grid or click the **Search** icon.  
Rows that match the keyword are filtered and displayed.

## Exporting to CSV

The exported CSV have this format:

- A header indicating the use of comma separator.
- A header with a list of the column names in the file.
- Each record is located on a separate line, delimited by a line break (LF).
- The last record in the file does not have a line break.

To export the data grid to a Comma-Separated Value file: 1.

- 1 Click the **Export to CSV** icon in the data grid toolbar.
- 2 A prompt is displayed to save the `data-export.csv` file on your local drive.
- 3 Save the file.

## Altering the results view

Results can have up to 100 rows and many columns. Investigating the data in the default size of the results panel can be cumbersome.

To make the view more suitable for exploring the data:

- 1 Drag the horizontal separator or click the Maximize icon in the data grid toolbar.  
When you click the Maximize icon, the screen shows these changes:
  - Results panel expands to cover larger part of the interface.
  - Data objects panel is hidden.
  - Query editor is hidden.
  - Toolbar tabs are not hidden, you can switch to a different tab or create a new one.
- 2 When the view is maximized, you can click the Minimize icon to return to the previous view.
- 3 To increase or decrease the row height, click the **More Actions** icon and select one of these options:
  - Short
  - Medium
  - Normal

## Error messages

Running a query can respond with errors.

The errors can be caused by either the user or by an ION response from the services utilized in the Compass user interface.

Different scenarios can respond with an error message, such as:

- Connection issues to service
- Query syntax error
- Invalid objects and properties
- Invalid use of functions
- Service errors

For more details on the error messages and scenarios, see the "Data Lake Queries" section in the *Infor ION Development Guide*.

When an error occurs, it is displayed in the Results panel with this structure:

- Error reason and error code
- Message describing the error
- Query ID that can be referenced for support
- UTC date and time when the error occurred

When the graphical interface fails to communicate with one of the backend services, such as Data Catalog or Compass backend service. The respective panels where the service is used, indicate the error state. If communication with Data Catalog cannot be achieved, the left Data Objects panel does not show any objects and displays a message.

If communication with the Compass backend service cannot be achieved, the results are not displayed, and the Result panel displays a message.

## Data Lake widgets

Use the Data Lake widgets, available from Homepages, to better understand the consumption and usage patterns of your Data Lake.

These statistical overviews for monitoring Data Lake metrics are accessible from the Statistics & Usage category within the Homepages Widget Catalog:

- Data Lake Ingestion by Object: measures the storage volume organized by data object name.
- Data Lake Ingestion over Time: measures the data object count and size ingestion rate to Data Lake over a selected period of time.
- Data Lake Storage Overview: provides an overview of the storage utilization and data object count in Data Lake.

See the "*Data Lake widgets*" section in the *Infor Ming.le Cloud Edition Online Help*.

## Chapter 17: ION Desk authorizations

Access to ION Desk can be restricted based on user's role.

The concepts behind the role based authorizations of ION Desk are described here.

This table shows the structure of the Authorizations menu:

Page	Usage
Desk Profiles	Create profiles to map user roles to permissions to access ION Desk pages.
Desk Permissions	View user-specific permissions to access ION Desk pages.
My Permissions	View permissions to access ION Desk pages of the current user.
Workflow Authorizations	Define mappings between user roles and workflows that may be started manually from Infor Ming.le.

### Resource

Each page that can be reached through the menu in ION Desk.

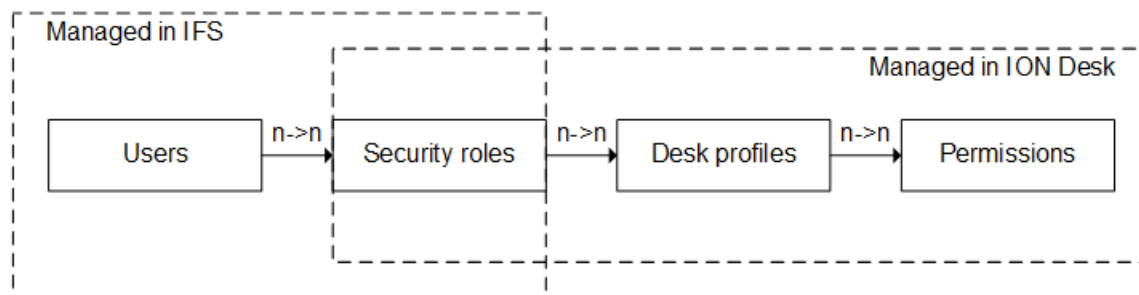
For example: **Connect > Data Flows** is a resource. Each resource can be authorized for no access or view-only access or full access.

### Permission

A combination of a resource and the allowed action on the resource. For example: Allowing view access on the **Data Flow** page is called 'Permission'.

### Desk Profile

A set of permissions. Each Desk Profile can be attached with or without roles. The relationship between various entities and the boundaries in which they exist is explained in this diagram:



## ION Desk users

Users and Security roles information needed for ION Desk authorizations are maintained in the Infor Federation Services (IFS) application. Note that, IFS allows users who belong to a Microsoft Active Directory only. Consequently, local Windows users cannot login to IFS or ION Desk. Business roles such as IT administrator or Support Engineer are also defined in IFS. Each user in IFS can have zero to many roles. In order to add users, security roles and to attach security roles to user. For more details see *Infor Federation Services Administration Guide (ifsag)*.

Users that require authorizations in ION Desk must meet these conditions:

- Configured as a user in IFS.
- The IFS Person field must be set.
- At least one IFS Security Role applicable for ION Desk must be assigned. You can assign a standard role or create your own roles. Later will be explained how to create your own security roles.

## Authorization framework

Resources that can be authorized usually correspond to a leaf level menu item. As a consequence the menu structure which every user sees depends on their permissions. If the user has at least view permission for a given resource. The required menu section and corresponding leaf level to access the page is displayed. If the user has no permissions for any resource in a menu section, the whole menu section is not displayed.

All leaf level menu items open a main access page from where a user can navigate to sub-pages. The level of access that is given to the user on the sub-pages is the same as the main page.

For example:

The **Data Flows** page under **Connect** is a resource. If the user has **view** level permissions for this resource, the **Connect > Data Flows** menu is shown to the user.

When the user clicks **Data Flows**, the main access page of **Data Flows** the list view of available data flows is displayed. The access control for the data flows details page is implicit. The access level on the details page is the same as the list page.

When the user can navigate to pages of another related resource, ION authorization framework expects that the user have explicit permissions for the destination resource. If not, the access channel to the related resource is either not shown or a `Permission Denied` error message is displayed. Same behavior is exhibited when importing one resource tries to create another related resource. In this case, the user is expected to have appropriate permissions on the related resource as well.

## Access level

Currently in ION Desk, these access levels are available:

- **None**  
This level denies the user from having any type of access on the resource.
- **View**  
This level provides read only access to the user on the resource. Typical actions associated with the 'View' access level are: Details, Search, Export
- **All**  
This level provides the user to run all actions on the resource. Typical actions associated with the 'All' access level are: All actions of View level, New, Save, Delete, Duplicate, Import, Activate and other custom actions that results in any change on ION Desk or ION Service.

The final decision on providing access to a resource is based on sum of all permissions a user may have. If the user has two ION Desk profiles attached to his role(s) and if a particular resource is denied in one profile while allowed in the other, the user will be given access to the resource.

## Standard IFS security roles and desk profiles

Every IFS installation comes with two standard security roles meant for the purpose of accessing ION Desk application. The roles are listed in this table:

Role Name	Description
IONDeskAdmin	Users with this role have full access to all ION Desk pages except the security admin pages
IONDeskSecurityAdmin	Users with this role have full access to the security admin pages of ION Desk

These standard security roles in IFS cannot be edited or deleted.

## Desk profile administration

Some Desk profiles are already linked to standard IFS security roles.

This table shows the Desk profiles which are delivered as standard with ION:

Desk profile name	Description	Security Roles
Viewer	Desk profile that provides view permission for all pages in ION Desk.	-

Desk profile name	Description	Security Roles
Modeler	Desk profile that provides all permissions for the modeling pages of the Connect and Process modules. Additionally it provides all permissions for these pages: <ul style="list-style-type: none"> <li>The <b>Codes</b> page under the <b>Configuration</b> menu section</li> <li>Document Authorization and Workflow authorization under the <b>Authorizations</b> menu section.</li> </ul>	IONDeskAdmin
Maintainer	Desk profile that provides all permissions for the management pages of the Connect and Process modules. Additionally it provides all permissions for these pages: <ul style="list-style-type: none"> <li>The pages under <b>Monitors &amp; Workflows &gt; Archive</b></li> <li>Logging, Error reporting, ION Service, Grid and Resource consumption pages under the <b>Configuration</b> menu section.</li> </ul>	IONDeskAdmin
Security Maintainer	Desk profile that provides all permission for Desk Profiles and Desk Permissions pages under the Authorization menu section.	IONDeskSecurityAdmin
BusinessRulesEditor	Desk profile that provides editor permissions for Business Rules. This profile must be used in combination with the security role DECISIONSERVICE-Editor.	DECISIONSERVICE-Editor
BusinessRulesApprover	Desk profile that provides approve permissions for Business Rules. This profile must be used in combination with the security role DECISIONSERVICE-Approver.	DECISIONSERVICE-Approver

The standard desk profiles in ION Desk cannot be deleted. Permissions definition within each standard desk profile cannot be edited. The default association of standard IFS security roles with each of these desk profiles cannot be edited. However new security roles can be added or removed.

## Adding a new IFS security role with a standard profile

- 1 Select **Authorizations > Desk Profiles..**  
The standard desk profiles delivered with ION are listed
- 2 Select a standard desk profile, for example: Modeler, and click **Details**.
- 3 In the Details page click the **Roles** tab.
- 4 Click **Add**  
A list of IFS security roles that can be attached with this profile will be displayed.
- 5 Select the role you want to add and click **OK**.
- 6 Click **Save**.

## Creating a desk profile

- 1 Click **Authorizations > Desk Profiles..**  
A list of existing profiles is displayed.
- 2 Click **Add** to open the details page.
- 3 Specify a Name and Description.  
The name must contain characters ranging from A-Z or a-z or 0-9 or an underscore (\_) or a dash (-).
- 4 On the **Permissions** tab, select the action for each resource available in the tree view.
- 5 In the Details page click the **Roles** tab.
- 6 Click **Add**  
A list of IFS security roles that can be attached with this profile is displayed.
- 7 Select the role you want to add and click **OK**.
- 8 Click **Save**.  
**Note:** All changes to desk profiles are effective immediately on the server side. But if a user is already logged in, the browser must be restarted for the new permissions to take effect.

## Duplicating a desk profile

Note that duplicating a desk profile only copies the permissions and not the roles. This is done to avoid disrupting the permissions of the user who are already part of the source profile.

For the rest duplicate works like other places in the ION Desk.

For Import/ Export of Desk profiles, see [Importing and exporting](#) on page 191.

## Modeling role-based access

You can specify a level of access to specific users.

- 1 Log on to the IFS application  
Ensure that an IFS security role exists corresponding to the role of the user whose access you want to control.  
Ensure that you added all the users in IFS, who want to access ION Desk.  
Ensure that appropriate IFS security roles are attached to the users.
- 2 Log on to the ION Desk application.
- 3 Examine if the standard ION Desk profiles provided out of the box are sufficient to access. If not create new ION Desk profiles with a set of permissions you prefer.
- 4 Attach the concerned IFS security role to your ION Desk profile.



## Examples

### Provide read only access of ION Desk to the quality control engineers

- 1 Log on to the IFS application.
- 2 Create the IFS security role **'Quality control engineer'**.
- 3 Add all users from the quality control department to IFS.
- 4 Link the IFS security role of quality control engineer with the users.
- 5 Log on to ION Desk.
- 6 The standard Desk profile of 'Viewer' already defines read only permissions for whole desk and it suffices your requirement.
- 7 Attach the IFS security role of **'Quality control engineer'** to the desk profile 'Viewer'  
Now all users who have the IFS security role **'Quality control engineer'** have read only access to the ION Desk.

### Provide Workflow modeling access to the business process admins

- 1 Log on to the IFS application.
- 2 Create the IFS security role **Business Process Administrator**.
- 3 Add all users who function as business process administrator to IFS.
- 4 Link the IFS security role of **Business Process Administrator** with the users.
- 5 Log on to ION Desk.
- 6 You find that no standard Desk profile provided give access to just the Workflow modeling pages.
- 7 Create a new desk profile called **workflow modeler**.
- 8 On the **Permissions** tab, select **All** access for resources under **Monitors & Workflows > Workflows**.
- 9 Attach the IFS security role of ' Business Process Administrator' to the desk profile Workflow modeler'.  
Now all users who have the IFS security role ' workflow modeler' have the read only access to the workflow modeling pages of ION Desk.

## Importing desk profiles

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Authorizations > Desk Profiles**.
- 3 Click **import**. A dialog box where you can locate and select the file opens.
- 4 Select the XML file to import.
- 5 Click **Open**. The desk profiles from the file are imported.  
If the file contains desk profiles that already exist, you are asked whether to skip or rename the imported items.

- Skip the already existing desk profiles.
- Rename the imported desk profiles.

## Exporting desk profiles

- 1 Open the menu by clicking the hamburger icon.
- 2 Select **Authorizations > Desk Profiles**.
- 3 Select the desk profiles to export.
- 4 Click **export**. A download dialog box of the browser is displayed.
- 5 Click **Save**.

The file is downloaded to the selected folder.

## Specific access behavior

This list contains some extra notes about special pages:

- The Home Page lists an overview of all active entities in the ION Service.
- If the BOD storage option is selected, then ION stores a copy of all BODs which can be viewed from OneView. A separate entry called 'BOD XML content' is added to the permissions tree. This is not a menu entry but still access control is applied separately. You must have **A11** permission for this resource if you must see the BOD content of messages displayed in ION.
- A few resources are not controlled for role based access. They are displayed when you log on as a valid Microsoft AD user. These resources are:
  - Home page.
  - My permissions page.
  - The **Reports** menu entry, available if you have generated audit reports for activities.

## Document authorization

### OneView documents

You can use OneView Documents if one of your IFS Security Roles has assigned a Desk Profile where the Permission **IONOneViewForDocuments** is set to **A11** or **View**.

A user authorized to use OneView Documents is by default authorized to see all documents. The **Document Content** permission is set to **All** by default.

Use IFS document authorizations, to limit the documents a user can see. You can limit document authorizations that are based on:

- Document type
- Accounting entity
- Location

To limit document authorizations:

- In IFS, ensure the Security Roles of the user have the Document authorizations configured.
- In ION Desk, ensure the Desk profiles of the user with the permission **Document Content** is set to **IFSManaged**.

To model the OneView document authorizations:

- 1 Log on to the IFS application.
  - Ensure that the user is configured as an IFS user.
  - Ensure that the user is assigned the appropriate security roles, accounting entities and locations.
  - Ensure that the security roles of the user have configured the appropriate document authorizations.

For more information about configuring this, see *Infor Federation Services Administration Guide (ifsag)*

- 2 Log on to the ION Desk application.
- 3 Select **Authorizations > Desk Profiles**.
- 4 Open the details for the profiles relevant for involved security roles.
- 5 Set the permission **Document Content** to **IFSManaged**.

If multiple security roles are linked to a single user, resulting in Document Content permissions of both **All**, **View** and **IFSManaged** the **All** permission prevails.

## Appendix A: Error Message fields

On the **Error BODs** page you can search on the Error Message or on the original message that caused the Error Message.

This table explains how the fields of the Confirm BOD are used by ION:

Fields	How is it determined
Tenant ID	Normalized message property
BOD ID	Normalized message property; If not available from /ConfirmBOD/ApplicationArea/BODID of message
Message ID	Normalized message property
From Logical ID	Normalized message property
Message	Normalized message Source
Creation dateTime	/ConfirmBOD/ApplicationArea/CreationDateTime of message
Original Message Id	/ConfirmBOD/DataArea/BOD/OriginalBOD/MessageHeader/MessageHeaderProperty/NameValue where attribute name=MessageId
Original Document ID	/ConfirmBOD/DataArea/BOD/OriginalApplicationArea/BODID of message
Original Document type	/ConfirmBOD/DataArea/BOD/OriginalBOD/MessageHeader/MessageHeaderProperty/NameValue where attribute name=BODType
Original From Logical Id	/ConfirmBOD/DataArea/BOD/OriginalBOD/MessageHeader/MessageHeaderProperty/NameValue where attribute name=FromLogicalId
Original To Logical ID	/ConfirmBOD/DataArea/BOD/OriginalBOD/MessageHeader/MessageHeaderProperty/NameValue where attribute name=ToLogicalId
Original Creation dateTime	/ConfirmBOD/DataArea/BOD/OriginalApplicationArea/CreationDateTime of message
Original Tenant ID	/ConfirmBOD/DataArea/BOD/OriginalBOD/MessageHeader/MessageHeaderProperty/NameValue where attribute name=TenantId

For Level, Severity, Error Type, Reason Code, and Reason this logic is used:

- If available: /ConfirmBOD/DataArea/BOD/BODFailureMessage/ErrorProcessMessage
  - Severity = Failure
  - Level = BOD
  - ReasonCode = ErrorProcessMessage/ReasonCode
  - Reason = ErrorProcessMessage/Description

- ErrorType = ErrorProcessMessage/Type
- else If available: /ConfirmBOD/DataArea/BOD/PartialBODFailureMessage/ErrorProcessMessage
  - Severity = Failure
  - Level = BOD-PARTIAL
  - ReasonCode = ErrorProcessMessage/ReasonCode
  - Reason = ErrorProcessMessage/Description
  - ErrorType = ErrorProcessMessage/Type

Do not to use the levels NOUN and NOUN-PARTIAL. Avoid using the level information in these XPATHs:

- /ConfirmBOD/DataArea/BOD/BODFailureMessage/NounFailureMessage/ErrorProcessMessage
- /ConfirmBOD/DataArea/BOD/PartialBODFailureMessage/NounFailureMessage/ErrorProcessMessage

## Appendix B: Standard functions for ION mappings

You can use several standard functions in ION mappings.

### Comparison

This table shows the comparison functions:

Function	Description	Example
Position	Returns the index position of the node that is currently being processed.	Example: <pre>//book[position()&lt;=3]</pre> Result: Selects the first three book elements.
Equals	Returns true if param1 and param2 are deep-equal to each other. Otherwise the function returns false.	
Not-Equals	Returns true if param1 and param2 are not deep-equal to each other. Otherwise the function returns false.	

### Constant

This table shows the constant functions:

Function	Description
String	Returns a string constant defined by the user.
Number	Returns a value constant defined by the user.

### Data type casting

This table shows the data type casting functions:

Function	Description	Example
Cast to string	Returns an entry as a string.	
Cast to Normalized String	Returns an entry as a normalized string. That is, Whitespace-replaced strings.	
Cast to Date and Time	Returns an entry as Date and time.	
Cast to Date	Returns an entry as Date.	
Cast to Time	Returns an entry as Time.	
Cast to Decimal	Returns an entry as Decimal.	
Cast to Integer	Returns an entry as Integer.	
Cast to Double	Returns an entry as Double.	
Cast to Float	Returns an entry as Float.	
Cast to Boolean	Returns an entry as Boolean.	
Cast to gYearMonth	Returns an entry as gYear-Month. That is, a period of one month.	Valid values include "2001-10", "2001-10+02:00", "2001-10Z", "2001-10+00:00", "-2001-10", or "-20000-04".
Cast to gYear	Returns an entry as gYear. That is, a period of one year.	Valid values include "2001", "2001+02:00", "2001Z", "2001+00:00", "-2001", and "-20000".
Cast to gMonthDay	Returns an entry as gMonth-Day. That is, a recurring period of time: yearly day.	Valid values are "--05-01", "--11-01Z", "--11-01+02:00", "--11-01-04:00", "--11-15", and "--02-29".
Cast to gDay	Returns an entry as gDay. That is, a recurring period of time: monthly day.	Valid values include "---01", "---01Z", "---01+02:00", "---01-04:00", "---15", and "---31".
Cast to gMonth	Returns an entry as gMonth. That is, a recurring period of time: yearly month.	Valid values include "--05", "--11Z", "--11+02:00", "--11-04:00", and "--02".
Cast to Long	Returns an entry as Long.	
Cast to Int	Returns an entry as Int.	
Cast to Short	Returns an entry as Short.	
Cast to Byte	Returns an entry as Byte.	
Cast to Token	Returns an entry as Token.	
Cast to Language	Returns an entry as Language.	

## Date and time

This table shows the date and time functions:

Function	Description
Adjust Date and Time to Time Zone	If the timezone argument is empty, this function returns a dateTime without timezone. Otherwise, this function returns a dateTime with the timezone.
Adjust Date to Time Zone	If the timezone argument is empty, this function returns a date without timezone. Otherwise, this function returns a date with the timezone.
Adjust Time to Time Zone	If the timezone argument is empty, this function returns a time without timezone. Otherwise, this function returns a time with the timezone.
Current Date and Time	Returns the current dateTime, with the timezone.
Current Date	Returns the current date, with the timezone..
Current Time	Returns the current time, with the timezone.
Format Date and Time - <code>date:format-date-time</code>	This function formats a date and time according to a pattern. The first argument specifies the date and time to be formatted. The second argument is a string that specifies the format pattern that is used to format the date-time.
Format Date - <code>date:format-date</code>	This function formats a date according to a pattern. The first argument specifies the date to be formatted. The second argument is a string that specifies the format pattern that is used to format the date. For example: <code>[M01]/[D01]/[Y0001]</code> ) = 09/19/2013
Format Time - <code>date:format-time</code>	This function formats a time according to a pattern. The first argument specifies the time to be formatted. The second argument is a string that specifies the format pattern that is used to format the time. For example: <code>[H01]:[m01] [z]'</code> ) = 09:26 GMT+10

## Logical

This table shows the logical functions:



Function	Description	Example
Choose	Used to choose one of several alternative outputs.	<pre>&lt;xsl:choose&gt;   &lt;xsl:when test="@cat='F'"&gt;Fic tion&lt;/xsl:when&gt;   &lt;xsl:when test="@cat='C'"&gt;Crime&lt;/xsl:when&gt;    &lt;xsl:when test="@cat='R'"&gt;Refer ence&lt;/xsl:when&gt;   &lt;xsl:otherwise&gt;Gener al&lt;/xsl:otherwise&gt; &lt;/xsl:choose&gt;</pre> <p>The first <code>xsl:when</code> element whose condition matches the current element in the source document is expanded. The others are ignored.</p>
If	The function takes a mandatory test attribute, whose value is a boolean expression. The contents of the <code>xsl:if</code> element are expanded only if the expression is true.	<p>Includes a hyperlink in the output only if the current element has a <code>preface</code> attribute:</p> <pre>&lt;xsl:if test="@preface"&gt;   &lt;a href="pref ace.html"&gt;Preface&lt;/a&gt; &lt;/xsl:if&gt;</pre>
Matches	Returns true if the string argument matches the pattern. Otherwise the function returns false.	<p>Example:</p> <pre>matches("Merano", "ran")</pre> <p>Result: true</p>
Not	The argument is first reduced to a boolean value by applying the <code>boolean()</code> function. Returns true if the boolean value is false. Returns false if the boolean value is true.	
TRUE	Returns the boolean value true.	
FALSE	Returns the boolean value false.	

## Math

This table shows the math functions:

Function	Description	Example
Highest*	Returns the nodes from the node set whose values are the maximum values in the node set. The highest* function inserts elements as an array and returns an array of elements.	
Lowest*	Returns the nodes from the node set whose values are the minimum values in the node set. The lowest* function inserts elements as an array and returns an array of elements.	
Maximum*	Returns the argument that is greater than the others. The maximum* function inserts elements as an array and returns one value.	Example:  <code>max ( (1, 2, 3) )</code>  Result: 3
Minimum*	Returns the argument that is less than the others. The minimum* function inserts elements as an array and returns one value. " "	Example:  <code>min ( (1, 2, 3) )</code>  Result: 1
Sum*	Returns the sum of the numeric values of nodes in the specified nodeset. The sum* function inserts elements as an array and returns one value.	
Add	Returns the sum of entry values.	
Subtract	Returns the difference of entry values.	
Multiply	Returns the multiplication product of entry values.	
Divide	Returns the quotient of entry values.	
String	Returns a string equivalent of the argument. The argument can be a number, boolean, or nodeset.	Example:  <code>string(314)</code>  Result: "314"

Function	Description	Example
Concatenate	Returns the result of concatenation of the specified strings.	<p>Example:</p> <pre>concat('XPath ', 'is', 'FUN!')</pre> <p>Result: 'XPath is FUN!'</p>
Join strings*	Returns a string that is created by concatenating the string arguments and using the sep argument as the separator. The join strings* function inserts separate strings as an array, uses the separator as a single string, and returns one string.	<p>Example:</p> <pre>stringjoin(('We', 'are', 'having', 'fun!'), ' ')</pre> <p>Result: 'We are having fun!'</p>
Length of string	Returns the length of the specified string. If there is no string argument, it returns the length of the string value of the current node.	<p>Example:</p> <pre>stringlength('Beatles')</pre> <p>Result: 7</p>
Substring	Returns the substring from the start position to the specified length. Index of the first character is 1. If length is not specified, the function returns the substring from the start position to the end.	<p>Example:</p> <pre>substring('Beatles', 1, 4)</pre> <p>Result: 'Beat'</p> <p>Example:</p> <pre>substring('Beatles', 2)</pre> <p>Result: 'eatles'</p>
Substring after	Returns the part of string 1 from where string 2 occurs in it until the end.	<p>Example:</p> <pre>substringafter('12/10', '/')</pre> <p>Result: '10'</p>
Substring before	Returns the part of string 1 from the start until string 2 occurs in it.	

Function	Description	Example
Tokenize*	Inserts two elements and returns elements as an array.	<pre>tokenize("XPath is fun", "\s+")</pre> Result: ("XPath", "is", "fun")
Normalize spaces	Removes extra whitespace from its argument string.	The argument string, with whitespace removed as follows: <ul style="list-style-type: none"><li>• All leading whitespace is removed.</li><li>• All trailing whitespace is removed.</li><li>• Within the string, any sequence of whitespace characters is replaced with a single space.</li></ul>

### Function templates

This table shows the function templates:

Function template	Description	Example
create-property	Returns a user-defined function that provides a property.	<pre data-bbox="1040 302 1393 827">&lt;xsl:element name=""Property""&gt;   &lt;xsl:element name=""NameValue""&gt;     &lt;xsl:attribute name=""name""&gt;       &lt;xsl:value-of se lect=""\$name""/&gt;     &lt;/xsl:attribute&gt;     &lt;xsl:attribute name=""type""&gt;       &lt;xsl:value-of se lect=""\$type""/&gt;     &lt;/xsl:attribute&gt;     &lt;xsl:value-of se lect=""\$element""/&gt;   &lt;/xsl:element&gt; &lt;/xsl:element&gt;</pre>
fill-property	Returns a user-defined function for filling a property.	<pre data-bbox="1040 882 1393 1323">&lt;xsl:element name=""NameValue""&gt;   &lt;xsl:attribute name=""name""&gt;     &lt;xsl:value-of se lect=""\$name""/&gt;   &lt;/xsl:attribute&gt;   &lt;xsl:attribute name=""type""&gt;     &lt;xsl:value-of se lect=""\$type""/&gt;   &lt;/xsl:attribute&gt;   &lt;xsl:value-of se lect=""\$element""/&gt; &lt;/xsl:element&gt;</pre>
get-first-item	Returns a user-defined function that provides the first item.	<pre data-bbox="1040 1373 1393 1491">&lt;xsl:copy-of se lect=""\$input-list[1]" copy-names paces=""no" /&gt;</pre>
get-first-value	Returns a user-defined function that provides the first value.	<pre data-bbox="1040 1541 1393 1629">&lt;xsl:value-of se lect=""\$input- list[1]" /&gt;</pre>
custom-function	Returns a user-defined function with blank function body.	
keep-date-reset-time	Returns a datetime string where the original input time is replaced by "00:00:00Z".	"2016-11-16T19:48:39.280Z" is converted into "2016-11-16T00:00:00Z".

Function template	Description	Example
keep-date-set-current-time	Returns a datetime string where the original input time is replaced by the current time.	
create-child-element	Creates a child element of the element to which the function is applied.	<pre data-bbox="1040 415 1393 621">&lt;xsl:element name="{ \$child-element-name }"&gt;   &lt;xsl:value-of select="\$child-element-value"/&gt; &lt;/xsl:element&gt;</pre>
create-child-element-with-attr	Creates a child element, with an attribute, of the element to which the function is applied.	<pre data-bbox="1040 672 1393 1087">&lt;xsl:element name="\$child-element-name"&gt;   &lt;xsl:attribute name="\$attribute-name"&gt;   &lt;xsl:value-of select="\$attribute-value"/&gt; &lt;/xsl:attribute&gt;   &lt;xsl:value-of select="\$child-element-value"/&gt; &lt;/xsl:element&gt;</pre>
get-elem-by-attr-name-and-value	<p data-bbox="610 1134 995 1224">Search for an element that has a specific attribute name and value:</p> <ul data-bbox="610 1234 995 1461" style="list-style-type: none"> <li data-bbox="610 1234 995 1360">• If the element has the searched attribute name and value, return the element.</li> <li data-bbox="610 1371 995 1461">• If the element does not have the searched attribute name and value, return null.</li> </ul> <p data-bbox="610 1472 995 1665">The first parameter always must be the actual source element, that is, node. This parameter cannot be a constant; it requires more than just an element name.</p>	
convert-numbers-to-words	Returns numeric values to text format displaying them as words. Upper limit is the billion and floats are rounded up or down accordingly	<pre data-bbox="1024 1686 1409 1749">&lt;xsl: number value="54" format="Ww" lang="en"/&gt;</pre>

**Note:** All of the above functions, except UDF, are not editable.

### User-defined functions

This table shows the user-defined functions:

Function	Description
<i>function name</i>	Returns a previously stored user-defined function.

## Appendix C: Tooltips for ION Mapper

Visual feedback is provided through tooltips with detailed messages. Here you can find an overview of the tooltip content and which tooltips are required.

### Tooltips

This table shows an overview of the tooltip content:

Occurrence	Tooltip (nodes)	Tooltip (elements)
[0..1]	Node(): Optional single-occurrence node ([0..1])	normalizedString: Optional single-occurrence element ([0..1])
[0.. ∞]	Node(): Optional multi-occurrence node ([0.. ∞])	normalizedString: Optional multi-occurrence element ([0.. ∞])
[1..1]	Node(): Mandatory single occurrence node ([1..1])	normalizedString: Mandatory single occurrence element ([1..1])
[1.. ∞]	Node(): Mandatory multi-occurrence node ([1.. ∞])	normalizedString: Mandatory multi-occurrence element ([1.. ∞])

This table shows the required tooltips:

Required tooltip (nodes)	Required tooltip (elements)	Required tooltip (attributes)
Node():Mandatory single-occurrence node	normalizedString: Optional single-occurrence element	normalizedString: Optional single-occurrence attribute