

Preparing for Insight Version 8.4

The following page describes the breaking changes that are introduced in Insight 8.4. In order to upgrade to the 8.4 version of Insight this document should be read in order to prepare to upgrade the Insight version.

For frequently asked questions about the upgrade to Insight 8.4, please see the [Insight 8.4 FAQs](#) page.

Page content:

- [Database changes](#)
- [New database map](#)
- [Event changes \(Automation\)](#)
- [Old reports and gadgets](#)
- [REST API](#)
 - [Objects CHANGED](#)
 - [Attachment CHANGED](#)
 - [General - Date and time request and response types](#)
 - [Date attribute type](#)
 - [Comparison between old and new](#)
 - [DateTime attributes](#)
 - [Comparison between old and new \(note timezone difference in the new response\)](#)
 - [Timestamps](#)
 - [Comparison between old and new](#)
- [Post functions](#)
 - [Assign Object based on Jira custom field - removed](#)
- [JQL Functions](#)
 - [attributeValue\(attribute, operator, value\) - removed](#)
- [JAVA API](#)
- [Facade](#)
 - [ObjectFacade](#)
 - [findObjectBeans](#)
 - [findObjectBeans](#)
 - [findObjectBeans](#)
 - [findObjectBeansByObjectSchema](#)
 - [findObjectBeansByName](#)
 - [findObjectBeansByAttributeValue](#)
 - [findObjectJiralssueBeans](#)
 - [findObjectJiralssueBeans](#)
 - [findObjectJiralssueBeans](#)
 - [ObjectSchemaFacade](#)
 - [loadObjectSchemaBean](#)
 - [ObjectTypeAttributeFacade](#)
 - [loadObjectTypeAttributeBean](#)
 - [loadObjectTypeAttributeBean](#)
 - [ObjectTypeFacade](#)
 - [loadObjectTypeBean](#)
 - [loadObjectTypeBean](#)
 - [ObjectTicketFacade \(New facade\)](#)
 - [findTicketsForObject](#)
- [Model classes](#)
 - [ObjectAttributeValueBean](#)
 - [RoleBean](#)
 - [ObjectBean](#)
- [Import](#)
 - [CSV import](#)

Database changes



Depending on the database type you might need to make manual changes. To make Insight more robust we have changed some database tables but we have seen issues in the Atlassian Active Object Framework during the upgrade process. This means that if you are using the below mentioned affected database types you need to follow the instructions to upgrade Insight.

Depending on the database type you are using you might have to follow below steps when upgrading to Insight 8.4.

SQL Server:

1. Before installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL DROP CONSTRAINT
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id;
```

2. Install new Insight version.
3. After installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL ADD CONSTRAINT
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id
FOREIGN KEY (OBJECT_ATTRIBUTE_ID) REFERENCES AO_8542F1_IFJ_OBJ_ATTR(ID);
```

MySQL:

1. Before installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL DROP FOREIGN KEY
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id;
```

2. Install new Insight version.
3. After installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL ADD CONSTRAINT
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id
FOREIGN KEY (OBJECT_ATTRIBUTE_ID) REFERENCES AO_8542F1_IFJ_OBJ_ATTR(ID);
```

H2:

1. Before installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL DROP FOREIGN KEY
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id;
```

2. Install new Insight version.
3. After installation, query the database:

```
ALTER TABLE AO_8542F1_IFJ_OBJ_ATTR_VAL ADD FOREIGN KEY
fk_ao_8542f1_ifj_obj_attr_val_object_attribute_id (OBJECT_ATTRIBUTE_ID)
REFERENCES AO_8542F1_IFJ_OBJ_ATTR(ID);
```

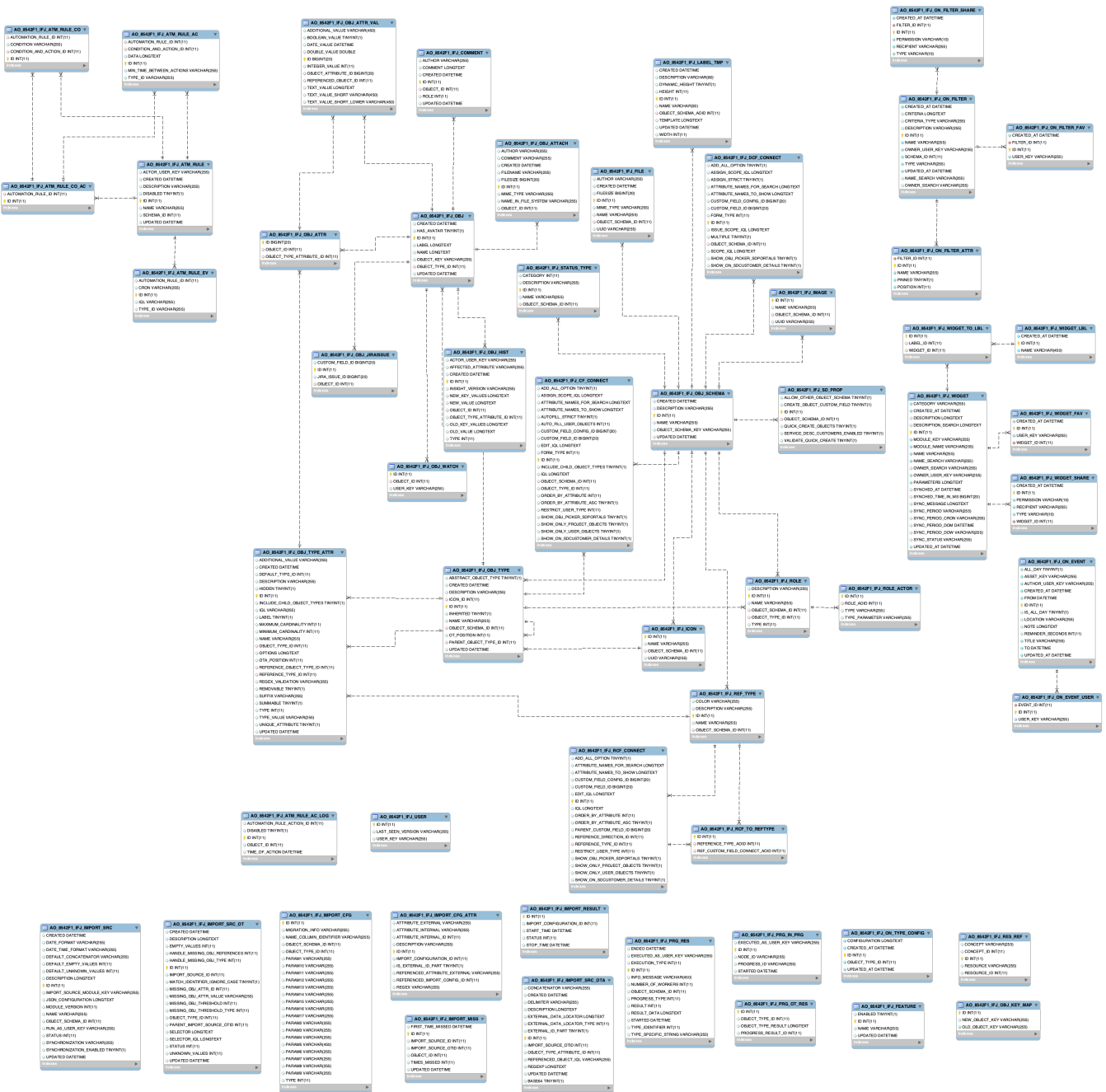
PostgreSQL & Oracle are not affected!



We recommend to re-index Insight after upgrade! This applies to all environments no matter database type!

New database map

The 8.4 release will affect the structure of our database. A digram of the the new database is shown below:



Event changes (Automation)

REMOVED

Object Cloned - The object clone event has been removed since it is a created event that happens.

Old reports and gadgets

REMOVED

All old reports and gadgets are removed

REST API

The following changes describes the changes done in the REST API. It is of importance if you are using a custom integration that consumes the Insight REST API. These are breaking changes for the Insight v1 API.

Objects CHANGED

Queries related to objects

Endpoint	Status	Replacement
GET /rest/insight/1.0/objecttype/{id}/objects	REPLACED	GET /rest/insight/1.0/iql/objects?iql=objectType={id}
GET /rest/insight/1.0/object/{id}/jiraissues	REPLACED	GET /rest/insight/1.0/objectconnectedtickets/{id}/tickets

Attachment CHANGED

Endpoint	Status	Replacement
GET /plugins/servlet/com.riadalabs.jira.plugins.insight/attachment/<attachment_id>/<filename>	REPLACED	GET /rest/insight/1.0/attachments/{attachmentId}

General - Date and time request and response types

In previous versions Insight REST API always returned all date and time information adjusted to the logged in users timezone as well as the configured date (or date time) format configured on the Jira server. To make the API more integration friendly all dates and times are changed to [ISO8601](#) format.

When mutating attribute values in Insight the input is expected to be sent in ISO8601 format as well. This change is made backward compatible meaning that the input will be assumed to be in ISO8601 format if the input fails to be parsed in this way the input will be parsed using the old way i.e. based on the Jira settings and the user timezone.

Date attribute type

All attribute values of type Date will be returned as an ISO8601 formatted date e.g. *2017-10-26*. The expected input when creating or updating is the same format.

Comparison between old and new

Old Response
<pre>"objectAttributeValues": [{ "displayValue": "29/Nov/2019", "value": "29/Nov/2019" }]</pre>
New Response
<pre>"objectAttributeValues": [{ "value": "2019-11-29", "searchValue": "2019-11-29", "displayValue": "29/Nov/2019", "referencedType": false }]</pre>

DateTime attributes

All date time attributes are now returned in ISO8601 format in UTC with milliseconds e.g. *2010-11-26T07:21:37.123Z*. Input is expected in ISO8601 format in an arbitrary timezone that is added as described by the standard e.g. *2010-11-26T06:21:37+0100*.

Comparison between old and new (note timezone difference in the new response)

Old Response

```
"objectAttributeValues": [
  {
    "displayValue": "26/Nov/19 9:07 AM",
    "value": "26/Nov/19 9:07 AM"
  }
]
```

New Response

```
"objectAttributeValues": [
  {
    "value": "2019-11-26T08:07:08.063Z",
    "searchValue": "2019-11-26T08:07:08.063Z",
    "displayValue": "26/Nov/19 9:07 AM",
    "referencedType": false
  }
]
```

Timestamps

All timestamps in Insight are now represented in the REST API in ISO8601 format in UTC. That is all created, updated, started, finished etc. are returned in ISO8601 format in UTC with milliseconds e.g. *2010-11-26T07:21:37.123Z*

Comparison between old and new

Old Response

```
"objectType": {
  "abstractObjectType": false,
  "created": "24/Oct/2019 01:11 AM",
  "icon": {
    "id": 29,
    "name": "Computer",
    "url16": "https://url/rest/insight/1.0/objecttype/143477/icon.png?size=16&inherited=true&abstract=false&time=1571919096176",
    "url48": "https://url/rest/insight/1.0/objecttype/143477/icon.png?size=48&inherited=true&abstract=false&time=1571919096176"
  },
  "id": 143477,
  "inherited": true,
  "name": "Laptop",
  "objectCount": 0,
  "objectSchemaId": 58583,
  "parentObjectType": 143476,
  "parentObjectTypeInherited": true,
  "position": 0,
  "type": 0,
  "updated": "24/Oct/2019 01:11 AM"
}
```

New Response

```
"objectType": {
  "id": 143477,
  "name": "Laptop",
  "type": 0,
  "icon": {
    "id": 29,
    "name": "Computer",
    "url16": "https://url/rest/insight/1.0/objecttype/143477/icon.png?size=16",
    "url48": "https://url/rest/insight/1.0/objecttype/143477/icon.png?size=48"
  },
  "position": 0,
  "created": "2019-10-24T00:11:02.891Z",
  "updated": "2019-10-24T00:11:02.891Z",
  "objectCount": 0,
  "objectSchemaId": 58583,
  "inherited": true,
  "abstractObjectType": false,
  "parentObjectTypeInherited": true
}
```

Post functions

Assign Object based on Jira custom field - removed

REMOVED

The post function Assign Object based on Jira custom field is now removed (it was deprecated in Insight 5.3). Please use the "Assign Objects from an IQL query based on data from an issue" post function instead.

JQL Functions

attributeValue(attribute, operator, value) - removed

REMOVED

The JQL function attributeValue("", "", "") has been removed and is replaced with the [IQLFunction](#) in JQL. The replacement will look similar to this

```
customField in iqlFunction("attribute operator value")
```

JAVA API


The JAVA API is used for the groovy scripting and custom integrations with Insight.

Facade


ObjectFacade

Name and status	Signature Insight <=8.3	Signature Insight 8.4
-----------------	-------------------------	-----------------------



findObjectBeans 	<pre>@Deprecated List<ObjectBean> findObjectBeans(int objectTypeId, boolean includeObjectTypesChilds, String filterName, int offset, Integer limit) throws InsightException;</pre>	<p>Use IQLFacade instead example IQL</p> <pre>label = "filterName" AND objectType IN (objectTypeId)</pre> <pre>label = "filterName" AND objectType IN objectTypeAndChildren(objectTypeId)</pre>
findObjectBeans 	<pre>@Deprecated List<ObjectBean> findObjectBeans(int objectTypeId, String iql, boolean includeObjectTypesChilds, String nameQuery, int offset, Integer limit) throws InsightException;</pre>	<p>Use IQLFacade instead example</p> <pre>label like "nameQuery" AND objectType IN (objectTypeId) AND iql</pre> <pre>label like "nameQuery" AND objectType IN objectTypeAndChildren (objectTypeId) AND iql</pre>
findObjectBeans 	<pre>@Deprecated List<ObjectBean> findObjectBeans(Set<Integer> objectTypeIds, String filterName, Long projectId, int offset, Integer limit) throws InsightException;</pre>	<p>Use IQLFacade instead example</p> <pre>objectType IN (objectTypeId1, objectTypeId2, ..., objectTypeIdN) AND label = "filterName"</pre>
findObjectBeansByObjectSchema 	<pre>@Deprecated List<ObjectBean> findObjectBeansByObjectSchema (int id, String namePrefix, int offset, Integer limit) throws InsightException;</pre>	<p>Use IQLFacade instead example</p> <pre>objectSchemaId = id AND label startsWith "namePrefix"</pre>
findObjectBeansByName 	<pre>@Deprecated List<ObjectBean> findObjectBeansByName(int objectTypeId, boolean includeObjectTypesChilds, String name) throws InsightException;</pre>	<p>Use IQLFacade instead example</p> <pre>label = "filterName" AND objectType IN (objectTypeId)</pre> <pre>label = "filterName" AND objectType IN objectTypeAndChildren(objectTypeId)</pre>
findObjectBeansByAttributeValue 	<pre>@Deprecated List<ObjectBean> findObjectBeansByAttributeValue(int objectTypeAttributeId, @Nonnull String operator, @Nonnull Object value) throws InsightException;</pre>	<p>This one requires the change of objectTypeAttributeId objectTypeAttributeName and value to be converted to it's string representation</p> <pre>objectTypeAttributeName = "valueAsString"</pre>

findObjectJiralssueBeans 	<pre>List<ObjectJiraIssueBean> findObjectJiraIssueBeans(int objectId) throws InsightException;</pre>	<pre>List<ObjectTicketConnectionEntry> findObjectTicketConnections(int objectId) throws InsightException;</pre>
findObjectJiralssueBeans 	<pre>List<ObjectJiraIssueBean> findObjectJiraIssueBeans(int objectId, int offset, int limit) throws InsightException;</pre>	<pre>@Deprecated List<ObjectTicketConnectionEntry> findObjectTicketConnections(int objectId, int offset, int limit) throws InsightException;</pre>
findObjectJiralssueBeans 	<pre>List<ObjectJiraIssueBean> findObjectJiraIssueBeans(Long jiraIssueId) throws InsightException;</pre>	<pre>List<ObjectTicketConnectionEntry> findObjectTicketConnections(long ticketId) throws InsightException;</pre>


ObjectSchemaFacade


Name and status	Signature Insight <=8.3	Signature Insight 8.4
loadObjectSchemaBean 	<pre>ObjectSchemaBean loadObjectSchemaBean(final int id) throws InsightException;</pre>	<pre>ObjectSchemaBean loadObjectSchema(final int id) throws InsightException;</pre>

ObjectTypeAttributeFacade


Deprecated	Signature Insight <=8.3	Signature Insight 8.4
loadObjectTypeAttributeBean 	<pre>ObjectSchemaBean loadObjectSchemaBean(final int id) throws InsightException;</pre>	<pre>ObjectTypeAttributeBean loadObjectTypeAttribute(final int id) throws InsightException;</pre>
loadObjectTypeAttributeBean 	<pre>ObjectTypeAttributeBean loadObjectTypeAttributeBean(int objectTypeId, String name) throws InsightException;</pre>	<pre>ObjectTypeAttributeBean loadObjectTypeAttribute(int objectTypeId, String name) throws InsightException;</pre>

ObjectTypeFacade

Deprecated	Signature Insight <=8.3	Signature Insight 8.4
loadObjectTypeBean 	<pre>ObjectTypeBean loadObjectTypeBean(final int id) throws InsightException;</pre>	<pre>ObjectTypeBean loadObjectType(final int id) throws InsightException;</pre>

loadObjectTyp eBean 	<pre>@Deprecated ObjectTypeBean loadObjectTypeBean(final int id, Object object) throws InsightException;</pre>	<pre>@Deprecated ObjectTypeBean loadObjectType(final int id, Object object) throws InsightException;</pre>
-------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------

ObjectTicketFacade (New facade)

Signature Insight 8.4	
findTicketsForO bject 	<pre>@NonNull ObjectTicketsEntry findTicketsForObject(int objectId, @Nullable Long filterId, @Nullable Integer limit);</pre>

Model classes

The model classes are used in some of the ObjectFacade methods as parameters and response objects.

ObjectAttributeValueBean

- Added getId() to retrieve the id
- Changed toString() from "[" + getValue() + "]" to "[" + id + "(" + getValue() + ")"]"

RoleBean

- getType() no longer return int but return a RoleType
- setType(int type) is changed to setType(RoleType)

ObjectBean

- getObjectAttributeBeans() only returns ObjectAttributeBean's which has a value.

Import

CSV import

The CSV import will now validate the content stream that the content is indeed a CSV file. This might affect import configurations that are importing CSV from a URL.