



INTEGRATION PACK FOR CA SERVICE DESK MANAGER

For Microsoft System Center Orchestrator

User Guide

Version 3.3

Kelverion Integration Pack for CA Service Desk Manager

Copyright 2013 Kelverion Inc. All rights reserved.

Published: April 2021

[Feedback](#)

Send suggestions and comments about this document to support@kelverion.com

Contents

Getting Started.....	5
System Requirements.....	5
Registering and Deploying the Integration Pack	5
Licensing the Integration Pack.....	6
Configuring the Integration Pack.....	7
CA Service Desk Manager Activities.....	8
Common Configuration Instructions for All Activities.....	8
Support CA Service Desk Object Types	8
Activity Properties.....	9
General Tab.....	9
Properties/Filters Tab	9
Run Behavior Tab.....	10
Published Data	11
Supporting Custom CA Service Desk Object Types	11
Object File XML Schema	12
ObjectTypes Element	12
ObjectType Element.....	12
Attribute Element	13
Close Ticket Activity	15
Create CI Activity.....	16
Create Attachment Activity.....	17
Create Object Activity	18
Get CI Details Activity.....	19
Get Objects Activity.....	20
Log Activity Activity.....	21
Monitor Objects Activity	22
Update Object Activity	23
Update CI Activity	24
Update Status Activity.....	25
Kelverion CA Service Desk Object Designer	26
CA Service Desk Object Type Designer	26

Working with Object Type Files	27
Working with Object Types.....	28
Working with Attributes	29

Getting Started

The Integration Pack for CA Service Desk Manager is an add-on for System Center Orchestrator that enables you to integrate with CA Service Desk Manager and automate service management processes, such as ticket creation, ticket monitoring and activity logging.

System Requirements

The Integration Pack for CA Service Desk Manager requires the following software to be installed and configured prior to implementing the integration. For more information about installing and configuring Orchestrator and CA Service Desk Manager, refer to the respective product documentation.

- Microsoft System Center Orchestrator *
- Microsoft .NET Framework 4.5.2
- CA Service Desk Manager 17.1, 17.2 or 17.3 with SOAP web services enabled.

* Please see kelverion.com/orchestrator for the latest Orchestrator support information.

Registering and Deploying the Integration Pack

After you download the integration pack file, you must register it with the Orchestrator management server and then deploy it to runbook servers and Runbook Designers.

To register the integration pack:

1. On the management server, copy the **.OIP** file for the integration pack to a local hard drive or network share.
2. Confirm that the file is not set to **Read Only** to prevent unregistering the integration pack later.
3. Start the **Deployment Manager**.
4. In the navigation pane of the Deployment Manager, expand **Orchestrator Management Server**, right-click **Integration Packs** to select **Register IP with the Management Server**. The **Integration Pack Registration Wizard** opens.
5. Click **Next**.
6. In the **Select Integration Packs or Hotfixes** dialog box, click **Add**.
7. Locate the **.OIP** file that you copied locally from step 1, click **Open** and then click **Next**.
8. In the **Completing the Integration Pack Wizard** dialog box, click **Finish**.
9. On the **End User Agreement** dialog box, read the Keverion License Terms, and then click **Accept**.
10. The **Log Entries** pane displays a confirmation message when the integration pack is successfully registered.

To deploy the integration pack:

1. In the navigation pane of the **Deployment Manager**, right-click **Integration Packs**, click **Deploy IP to Runbook Server or Runbook Designer**.
2. Select the integration pack that you want to deploy, and then click **Next**.
3. Enter the name of the runbook server or computers with the Runbook Designer installed, on which you want to deploy the integration pack, click **Add**, and then click **Next**.
4. Continue to add additional runbook servers and computers running the Runbook Designer, on which you want to deploy the integration pack. Click **Next**.
5. In the **Installation Options** dialog box, configure the following settings.
6. To choose a time to deploy the integration pack, select the **Schedule installation** check box, and then select the time and date from the **Perform installation** list.
7. Click one of the following:
 - a. **Stop all running runbooks before installing the integration pack** to stop all running runbooks before deploying the integration pack.
 - b. **Install the Integration Packs without stopping the running Runbooks** to install the integration pack without stopping any running runbooks.
8. Click **Next**.
9. In the **Completing Integration Pack Deployment Wizard** dialog box, Click **Finish**.
10. When the integration pack is deployed, the **Log Entries** pane displays a confirmation message.

For more information about how to install integration packs, see the [How to Install an Integration Pack](https://technet.microsoft.com/en-us/library/hh420346.aspx) (<https://technet.microsoft.com/en-us/library/hh420346.aspx>).

Important: If you make changes to **ObjectTypes** file then you should rename it to something other than ObjectTypes.xml or change the default location so that it is not lost or overwritten when you uninstall the integration pack or deploy an updated version.

Licensing the Integration Pack

After you register and deploy the integration pack you must provide a valid Keverion license before running any runbooks that contain activities from the integration pack

To deploy the integration pack license file:

1. Copy the .KAL license file to %PROGRAMFILES(X86)%\Keverion Automation\Licenses
2. Repeat for each Orchestrator Runbook Server and Runbook Designer host system.

Important: If you are upgrading an existing deployment of this integration pack that uses version 2.1 or earlier, you must contact your sales contact or info@kelverion.com to obtain a new license before upgrading to this version.

Configuring the Integration Pack

A connection establishes a reusable link between Orchestrator and a CA Service Desk Manager server. You can create as many connections as you require specifying links to multiple Service Desk servers. You can also create multiple connections to the same table to allow for differences in security permissions for different user accounts.

To set up a CA Service Desk Manager configuration:

1. In the Client, click the **Options** menu, and select *KA CA Service Desk Manager*. The **KA CA Service Desk Manager** dialog box appears.
2. On the **Configurations** tab, click **Add** to begin the configuration setup. The **Add Configuration** dialog box appears.
3. In the **Name** box, enter a name for the configuration. This could be the name of the Service Desk server or a descriptive name to distinguish the type of configuration.
4. Click the ellipsis button (...) next to the **Type** box and select *CA Service Desk*.
5. In the **Server URL** box, type the name or URL of the CA Service Desk web service. For example: `http://172.16.254.1:8080/axis/services/USD_R11_WebService`. Note that the path portion following the port number is only required if the path to your CA Service Desk SOAP web service is different than `/axis/services/USD_R11_WebService`. Secure Example: `https://casd.company.com:8443`
6. In the **User Name** and **Password** boxes, type the credentials that Orchestrator will use to connect to the CA Service Desk Manager server.
7. Optionally, in the **Object Types File** box, select the path to the file used to define the object types present in your CA Service Desk Manager environment. **Important:** If you make changes to this file then rename it to something other than `ObjectTypes.xml` or change the default location so that it is not lost or overwritten when you uninstall the integration pack or deploy an updated version.
8. Add additional connections if applicable
9. Click **OK** to close the configuration dialog box, and then click **Finish**.

The default **Object Types File** defines core CA Service Desk ticket object types and their immediate dependencies. If the default **objecttypes.xml** file is not representative of the object types in your CA Service Desk environment, you can customize this file as required or create a new file which you can select when you define your connection(s) to CA Service Desk.

All integration pack activities that use a given configuration will share the same object type definitions. After making changes to the object types file you should review and save each integration pack activity that is dependent on the configuration to ensure that the new object type

definitions are integrated into your runbooks. For more information on how to customize the object types file refer to **Supporting Custom CA Service Desk Object Types**.

CA Service Desk Manager Activities

This integration pack adds the KA CA Service Desk Manager category to the **Activities** pane in the Client. This category contains the following activities:

- Close Ticket
- Create CI
- Create Attachment
- Create Object
- Get CI Details
- Get Objects
- Log Activity
- Monitor Objects
- Update Object
- Update CI
- Update Status

Common Configuration Instructions for All Activities

The following configuration instructions apply to all activities in this integration pack. Links to this section are included in the configuration instructions for each activity.

Support CA Service Desk Object Types

The integration pack supports most core CA Service Desk object types and their immediate dependencies. These include:

- Activity Log
- Announcement
- Change Order
- Configuration Item*
- Request
- Contact
- Department
- Group
- Incident
- Issue
- Location
- Organization
- Problem
- Site

Note: Support for Configuration Items is limited to generic attributes and confined to the Get/Monitor Object and Update Object activities.

Activity Properties

Each activity has a set of required or optional properties that define the configuration of that activity. This includes how it connects to other activity or how the activity performs its actions. You can view or modify activity properties in the Orchestrator Client.

Configure the properties for an activity:

1. Double-click the activity. Alternatively, you can right-click the activity, and then click **Properties**.
2. To save your configuration entries, click **Finish**.

In the activity properties dialog box, several tabs along the left side provide access to general and specific settings for the activity. Although the number of available tabs for activity properties differs from activity to activity, all activities will have a **General** tab, a **Properties** tab and/or **Filters** tab, and a **Run Behavior** tab. Some activities may have additional tabs.

General Tab

This tab contains the **Name** and **Description** properties for the activity. By default, the **Name** of the activity is the same as its activity type, and the **Description** is blank. You can modify these properties to create more descriptive names or provide detailed descriptions of the actions of the activity.

Properties/Filters Tab

These tabs contain properties that are specific to the activity.

All activities in this integration pack have the **Configuration Name** property at the top of the **Properties** tab. This property is used to specify the connection to a CA Service Desk Manager server.

To configure the Configuration Name property:

1. Click the ellipsis (...) button next to the **Name** field, and then select the applicable connection name. Connections displayed in the list have been previously configured as described in [Configuring the CA Service Desk Manager Connections](#).

Filter Behavior

The Monitor and Get activities use filters to determine the values that will invoke a runbook or retrieve activities. Property values of potential candidates are compared to the values of the filters to determine if they meet the criteria. When matching against values, you select one of the available methods of comparison. An option is provided to either match or not match the filter using each method. For example, the "Does not" version of a method causes alerts that do not match the filter to trigger the runbook.

- **Equals:** the field of the record exactly matches the text or number specified in the filter.
- **Does not equal:** the field of the record does not exactly match the text or number specified in the filter.
- **Is less than:** the field of the record is less than the number specified in the filter.

- **Is less than or equal to:** the field of the record is less than or equal to the number specified in the filter.
- **Is greater than:** the field of the record is greater than the number specified in the filter.
- **Is greater than or equal to:** the field of the record is greater than or equal to the number specified in the filter. The wildcard characters, '%' and '_' can be used to match multiple characters or a single character, respectively.
- **Matches:** use wildcards to specify a pattern that the text must match. The two wildcard values are the percent (%) and the underscore (_). The percent will match any number of characters, while the underscore will only match a single character.
- **Does not match:** use wildcards to specify a pattern that the text does not match. The two wildcard values are the percent (%) and the underscore (_). The percent will match any number of characters, while the underscore will only match a single character.

Run Behavior Tab

This tab contains the properties that determine how the activity handles multi-value published data and what notifications will be sent if the activity fails or runs for an excessive period of time.

Multi-Value Published Data Behavior

The Get activities retrieve information from another activity or outside source and can return one or more values in the published data. For example, when you use the Get Collection Member activity, the data output from that activity might be a list of computers that belong to the specified collection.

By default, the data from the Get activity will be passed on as multiple individual outputs. This invokes the next activity as many times as there are items in the output. Alternatively, you can provide a single output for the activity by enabling the **Flatten** option. When you enable this option, you also choose a formatting option:

- **Separate with line breaks.** Each item is on a new line. This format is useful for creating human-readable text files for the output.
- **Separate with _.** Each item is separated by one or more characters of your choice.
- **Use CSV format.** All items are in CSV (comma-separated value) format. This format is useful for importing data into spreadsheets or other applications.

The activity will produce a new set of data every time it runs. The **Flatten** feature does not flatten data across multiple instances of the same activity.

Event Notifications

Some activities are expected to take a limited amount of time to complete. If they do not complete within that time they may be stalled or there may be another issue preventing them from completing. You can define the number of seconds to wait for completion of the action. After this period a platform event will be sent, and the issue will be reported. You can also choose whether to generate a platform event if the activity returns a failure.

Be notified when the activity takes longer than a specified time to run or fails to run:

1. In the **Event Notifications** box, enter the **number of seconds** of run time before a notification is generated.
2. Select **Report if activity fails to run** to generate run failure notifications.

For more information about Orchestrator events, see the “Event Notifications ” topics in the [Runbook Properties](https://technet.microsoft.com/en-us/library/hh489610.aspx#EventNotifications) (https://technet.microsoft.com/en-us/library/hh489610.aspx#Event Notifications).

Published Data

Published data is the foundation of a working runbook. It is the data produced as a result of the actions of an activity. This data is published to an internal data bus that is unique for each runbook. Subsequent activities in the runbook can subscribe to this data and use it in their configuration. Link conditions also use this information to add decision-making capabilities to runbooks.

An activity can only subscribe to data from the activities that are linked before it in the runbook. You can use published data to automatically populate the property values needed by activities.

To use published data:

1. Right-click the property value box, click **Subscribe**, and then click **Published Data**.
2. Click the **Activity** drop-down box and select the activity from which you want to obtain the data.
3. To view additional data elements common to all activities, select **Show Common Published Data**.
4. Click the published data element that you want to use, and then click **OK**.

For a list of the data elements published by each activity, see the Published Data tables in the activity topic. For information about the common published data items, see the [Published Data](http://technet.microsoft.com/en-us/library/hh403821.aspx) (http://technet.microsoft.com/en-us/library/hh403821.aspx).

Supporting Custom CA Service Desk Object Types

The Integration Pack for CA Service Desk uses an external XML file to define which CA Service Desk object types are available when configuring the activities in the Integration Pack. By default, the set of available object types includes the core CA Service Desk ticket objects, including Issue, Request, Change Order, Problem and Incident.

Starting with version 1.1, the path to the object types file can be specified in the CA Service Desk Configuration dialog. This change makes it easier for users to customize the object types that are available in their Service Desk environment and to support multiple object types collections in multi-tenant environments.

The Object Types file is an XML formatted file that is used to map CA Service Desk object types to Microsoft System Center Orchestrator. The following sections provide descriptions of the elements in the Object Type schema.

Object File XML Schema

The following outlines the XML schema used by the **ObjectTypes** file to define the object types in your CA Service Desk Manager environment.

ObjectTypes Element

The **<objectTypes>** element is the top-level element and is used to contain the collection of **<objectType>** elements that define the object types in your CA Service Desk Manager environment.

Syntax

```
<objecttypes>

    ... object type definitions ...

</objecttypes>
```

Parent

None.

Attributes

None.

ObjectType Element

The **<objectType>** element defines how a specific object type appears in System Center Orchestrator and how it integrates with CA Service Desk Manager.

Syntax

```
<objectType id="name"

    relAttr="attribute id"

    commonName="attribute id"

    creationDate="attribute id"

    modifiedDate="attribute id">

    ... display name and attributes ...

</objectType>
```

Attributes

The **<objecttype>** element has the following attributes.

Attribute	Description
Id	The name of the object type as defined in the CA Service Desk object definition.
relAttr	Identifies the attribute that will represent this object type when it is referenced (used as an SREL) by another object.
commonName	Identified the attribute to be displayed in browser lists or published data.
creationDate	Identifies the attribute used to indicate the date/time that an instance of the object type was created.
modifiedDate	Identifies the attribute used to indicate the date/time that an instance of the object type was last modified.

DisplayName Element

The **<displayName>** element specifies the name used to display the object type in System Center Orchestrator.

Syntax

```
<displayName>
```

```
    <!--The name used to display the object type in Orchestrator -->
```

```
</displayname> id="name"
```

Parent

The **<objecttype>** element.

Attributes

None.

Attribute Element

The **<attribute>** element defines how specific object type attribute will appear in System Center Orchestrator and how it integrates with CA Service Desk Manager

Syntax

```
<attribute id="name"
```

```
    dataType="integer|double|string|duration|uuid|date"
```

```
    required="true|false"
```

```
    readonly="true|false"
```

```
    lookup="true|false"
```

```
    default="value">
```

```
    <!-- Display Name -->
```

```
</attribute>
```

Parent

The <objecttype> element.

Children

None.

Attributes

The <attribute> element has the following attributes.

Attribute	Description
Id	The name of the attribute as defined in the CA Service Desk object definition.
dataType	Identifies the data type of the attribute's value. Supported data types include: <ul style="list-style-type: none">• Integer• double• string• duration• uuid• date
required	Indicates whether the attribute is required when creating new instance of the object type.
readonly	Indicates whether the attribute is read only.
lookup	Indicates whether the Integration Pack activities should provide browsers to assist users in selecting values for the attribute. Users should consider the performance impact of enabling lookup for an attribute, especially when the attribute references another object type that contains many instances.
default	The default value that should be used to initialize the attribute in System Center Orchestrator. Note that this default value only applies to System Center Orchestrator and does not override the default value that may have been configured in CA Service Desk's schema designer.

Close Ticket Activity

The **Close Ticket** activity is used in a runbook to close a ticket in CA Service Desk Manager. This activity generates a CA Service Desk activity log.

Required Properties

You must configure the following properties.

Description	Describes why the ticket was closed.
Object Handle	The unique object handle of the ticket to close.

Published Data

This activity publishes the following activity-specific data items.

Activity Log ID	The ID of the activity log entry that was created.
Description	Describes why the ticket was closed.
Object Handle	The unique object handle of the ticket to close.

Create CI Activity

The **Create CI** activity is used in a runbook to create a new Service Desk asset or Configuration Item (CI) object.

Important: If a CI exists with the name that was specified when you defined the Create CI activity, then a new CI **will not be created** and the existing CI will be updated.

Required Properties

You must configure the following properties. This activity may provide additional properties that you must configure, depending on your **Class** selection.

Class	The class of CI object that you want to create
--------------	--

Optional Properties

This activity provides different optional properties, depending on your **Class** selection, and you can configure them as required to initialize the new CI.

Published Data

This activity publishes the following activity-specific data items.

Extension Handle	The unique handle of the extension object used to hold detailed information about the CI object
Object Handle	The unique handle of the CA Service Desk Manager object that was created.

Create Attachment Activity

The **Create Attachment** activity is used in a runbook to upload and attach a file to a specified CA Service Desk object.

Required Properties

You must configure the following properties.

Content Type	Identifies the type of document
Description	Describes the document
Document Repository	The name of the document repository that will contain the uploaded document.
File Path	The full path to the file to upload
Object Handle	The handle of the object that the file will be attached to.

Published Data

This activity publishes the following activity-specific data items.

Attachment Handle	The object handle of the attachment that was created.
Content Type	Identifies the type of document
Description	Describes the document
Document Repository	The name of the document repository that will contains the uploaded document.
File Path	The full path to the file that was uploaded
Object Handle	The handle of the object that the file is attached to.

Create Object Activity

The **Create Object** activity is used in a runbook to create a new Service Desk object.

Required Properties

You must configure the following properties. This activity may provide additional properties that you must configure, depending on your **Object Type** selection.

Object Type	Indicates the type of Service Desk object to create.
--------------------	--

Optional Properties

This activity provides different optional properties, depending on your **Object Type** selection, and you can configure them as required to initialize the new object.

Published Data

This activity publishes the following activity-specific data items.

ID	The unique id of the CA Service Desk Manager object that was created.
Object Handle	The unique handle of the CA Service Desk Manager object that was created.
Object Type	Indicates the type of CA Service Desk object that was created.

Get CI Details Activity

The **Get CI Details** activity is used in a runbook to retrieve detailed information about a specific CA Service Desk CI object.

Required Properties

You must configure the following properties.

Family	A CA Service Desk CI family
Object Handle	The unique handle of the CA Service Desk CI object to retrieve

Published Data

This activity publishes the following activity-specific data items. Additional published data will be provided, depending on your **Family** selection.

Object Handle	The unique handle of the CA Service Desk Manager object
----------------------	---

Get Objects Activity

The **Get Objects** activity retrieves objects from CA Service Desk Manager that match filter criteria that you specify.

Required Properties

You must configure the following properties.

Object Type	Indicates the type of CA Service Desk objects to retrieve.
--------------------	--

Filters

This activity provides the following filters for all object types. This activity provides additional filters, depending on your **Object Type** selection. You can combine one or more filters to selectively retrieve only those objects that are of interest.

ID	The unique id of the CA Service Desk Manager object. This could be an integer or UUID.
Object Handle	The unique handle of the CA Service Desk Manager object

Published Data

This activity publishes the following activity-specific data items. Additional published data will be provided based on your **Object Type** selection.

ID	The unique id of the CA Service Desk Manager object.
Object Handle	The unique handle of the CA Service Desk Manager object
Object Type	The name of the CA Service Desk object type to query.

Log Activity Activity

The **Log Activity** activity is used in a runbook to create an activity log for an Issue, Request, Incident, Problem or Change Order.

Required Properties

You must configure the following properties.

Activity Type	The type of activity log to create. Options include Callback, Log Comment and Research.
Description	A comment or description of the activity.
Object Handle	The unique object handle of a Service Desk ticket.

Optional Properties

This activity provides the following optional properties that you can configure, as required.

Internal	Indicates whether the activity log is internal
Time Spent	The time spent on the activity. The value must be in the format [d.]hh:mm:ss.

Published Data

This activity publishes the following activity-specific data items.

Activity Log ID	The ID of the activity log entry that was created.
Activity Type	The type of activity log that was created. Possible values include Callback, Log Comment and Research.
Description	A comment or description of the activity.
Internal	Indicates whether the activity log is internal
Object Handle	The unique object handle of the ticket to escalate.
Time Spent	The time spent on the activity in the format [d.]hh:mm:ss.

Monitor Objects Activity

The **Monitor Objects** activity is used in a runbook to monitor Service Desk for new or modified objects using filter criteria that you specify.

Tip: Only CA Service Desk object types that have the required create and modified date attributes are available for monitoring.

Required Properties

You must configure the following properties.

Object Type	The name of the CA Service Desk object type to monitor.
Monitor Interval	The interval in seconds that the monitor waits each time it polls CA Service Desk for changes.

Filters

This activity provides the following filters for all object types. This activity provides additional filters, depending on your **Object Type** selection. You can combine one or more filters to selectively control which objects will trigger the monitor.

ID	The unique id of the CA Service Desk Manager object.
Object Handle	The unique handle of the CA Service Desk Manager object

Published Data

This activity publishes the following activity-specific data items. Additional published data will be provided based on your **Object Type** selection.

ID	The unique id of the CA Service Desk Manager object. This could be an integer or UUID.
Object Handle	The unique handle of the CA Service Desk Manager object
Object Type	The name of the CA Service Desk object type to query.

Update Object Activity

The **Update Object** activity is used in a runbook to update one or more attributes of a specified Service Desk object.

Required Properties

You must configure the following properties.

Object Handle	The unique handle of the Service Desk object to update.
Object Type	Indicates the type of Service Desk object to update.

Optional Properties

This activity provides different optional properties, depending on your **Object Type** selection, and you can configure them as required to update the object.

Published Data

This activity publishes the following activity-specific data items.

ID	The unique id of the CA Service Desk Manager object that was updated.
Object Handle	The unique handle of the CA Service Desk Manager object that was updated.
Object Type	Indicates the type of Service Desk object that was update.

Update CI Activity

The **Update CI** activity is used in a runbook to update an existing Service Desk asset or CI object instance.

Tip: When making changes to either the **CI Asset** (is_asset) or **Active** (delete_flag) properties, you should not include changes to any other properties.

Required Properties

You must configure the following properties.

Class	The class of CI object that you want to create
Object Handle	The unique handle of the object to be updated

Optional Properties

This activity provides different optional properties, depending on your **Class** selection, and you can configure them as required to update the CI.

Published Data

This activity publishes the following activity-specific data items.

Object Handle	The unique handle of the CA Service Desk Manager object that was created.
Extension Handle	The unique handle of the extension object used to hold detailed information about the CI object

Update Status Activity

The **Update Status** activity is used in a runbook to update the status of a Service Desk Issue, Request, Incident, Problem or Change Order. This activity generates a CA Service Desk activity log.

Required Properties

You must configure the following properties.

Description	Describes why the status of the ticket is being changed.
Object Handle	The unique handle of the Service Desk object to update.
Object Type	Indicates the type of Service Desk object to update.
Status	The new status to assign to the ticket.

Published Data

This activity publishes the following activity-specific data items.

Activity Log ID	The ID of the activity log entry that was created.
Description	Describes why the status of the ticket is being changed.
ID	The unique id of the CA Service Desk Manager object that was updated.
Object Handle	The unique handle of the CA Service Desk object that was updated.
Object Type	Indicates the type of Service Desk object that was update.
Status	The new status to assigned to the ticket.

Kelverion CA Service Desk Object Designer

The Kelverion Integration Pack for CA Service Desk Manager uses a SOAP Web Service API to facilitate integration with your CA Service Desk environment. Communication with CA Service Desk Manager using this API is carried out at a very low-level and incorporates names and identifiers that are very different from what you would see when interacting with CA Service Desk Manager using a web browser.

To provide runbook authors with a familiar interface for working with CA Service Desk, the Kelverion Integration Pack for CA Service Desk Manager incorporates a configurable **ObjectTypes file** to store information about the object types in your CA Service Desk environment.

The ObjectTypes file is used to:

- Define the primary object types that are available to the activities in the integration pack.
- Define secondary objects types to support the collection of primary object types that you can integrate with. While you cannot integrate with these object types directly, they are used to provide browsers for various properties.
- Define how the object types will be displayed and function in Orchestrator and how information from Orchestrator maps to the CA Service Desk SOAP Web Service API.

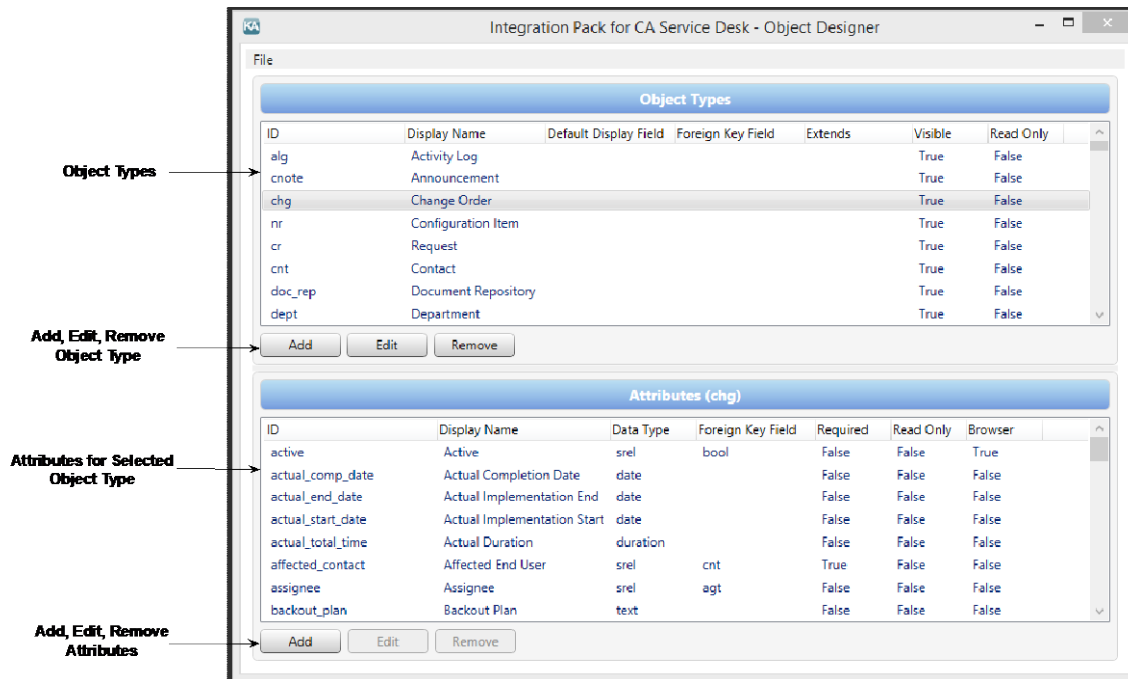
The Kelverion Integration Pack for CA Service Desk is automatically deployed with a default ObjectTypes file, which is located at *COMMONPROGRAMFILES(X86)\ \Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\482a59fe-1da7-4ec9-8a7c-b759dfaf07a7*. The object types that are contained in the default ObjectTypes.xml file is compatible with the object types in a standard ITIL deployment of CA Service Desk Manager 12.6.

For the Kelverion Integration Pack for CA Service Desk to integrate properly with your CA Service Desk environment it may be necessary to make changes to the ObjectTypes file. In order to make modifications to this file easier and more manageable, we have included a new **Object Types Designer**, which provides a graphical interface for managing Object Type files.

CA Service Desk Object Type Designer

The Kelverion CA Service Desk Object Designer is a Windows application that you can use to manage object types that the Kelverion Integration for CA Service Desk Manager can integrate with. **Any changes made in the Object Type Designer only affect the ObjectTypes file used to manage integration between CA Service Desk and System Center Orchestrator and do not change, in any way, the configuration of your CA Service Desk server and the object types that it supports.**

The following diagram illustrates the Kelverion CA Service Desk Object Type Designer. The top list view is used to display the object types that the integration pack can integrate with, while the bottom list view is used to display the attributes contained in the object type that is currently selected.



Working with Object Type Files

Opening an Existing ObjectTypes file

You can open an existing ObjectTypes file for viewing or editing. Note that the default ObjectTypes file that is deployed with the integration pack is located at *COMMONPROGRAMFILES(X86)\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\482a59fe-1da7-4ec9-8a7c-b759dfaf07a7\ObjectTypes.xml* file

1. Open the Windows Start Menu and select **Kelverion Automation/CA Service Desk Object Designer**.
2. Click the **File** menu and click **Open**. Use the Open File dialog to locate the Object Types XML file that you want to edit and click **Open**. The Object Types list should display the collection of object types that are defined in the ObjectTypes file that you selected.

Saving Your Changes

In order to ensure that your changes are not lost you must ensure that you save them prior to closing the Object Designer. Also, as a best practice we recommend saving your changes to a file other than the default *COMMONPROGRAMFILES(X86)\Microsoft System Center 2012\Orchestrator\Extensions\Support\Integration Toolkit\482a59fe-1da7-4ec9-8a7c-b759dfaf07a7\ObjectTypes.xml* file. Failure to do so will result in your changes being lost when you uninstall the integration pack or deploy a new version.

1. To save any changes that you have made, click the **File** menu, and then click **Save**.
2. Alternatively, you can save your changes to another file. To do so, click the **File** menu and then click **Save As**.

Working with Object Types

The Object Type designer lets you change the follow object type details:

- **ID:** specifies the identifier used to uniquely identify the object type.
- **Display Name:** specifies the name used to display the object type in Orchestrator.
- **Default Display Attribute:** specifies the attribute that is referenced when building a list of values to represent instances of the object type.
- **Foreign Key Attribute:** specifies the attribute that is referenced when creating a foreign key association (SREL) to the object type.
- **Modified Date Attribute:** specifies the date/time attribute that is referenced by the Monitor Objects activity to determine whether an instance of the object type has been modified.
- **Extends:** specifies object type that the object type extends. Used when creating new configuration item types to extend the base Configuration Item (nr) type.
- **Visible:** specifies whether the object type is visible in the Object Type browser when configuring and integration pack activity.
- **Read Only:** specifies whether instances of the object type can be changed.

Add a new object type:

Click the **Add** button underneath the **Object Type List**. An empty **Object Type Details** dialog should be displayed.

1. Specify details of your new object type and click **OK**. The new object type should appear in the **Object Type List**.
2. Add attributes to your new object type as described in the Working with Attributes section.

Make a change to an existing object type:

1. Select the object type that you want to change and click the **Edit** button located underneath the object type list. The **Object Type Details** dialog should appear. Click **Cancel** to disregard any changes that you have made.
1. Make the desired changes and click **OK**.

Removing an object type

You can remove one or more object types, although care should be taken when doing so as not to break the foreign key (SREL) relationships between the various object types in your CA Service Desk environment.

Remove an object type:

1. Select the object types you want to remove and click the **Remove** button just below the Object Type List.
2. When you are prompted, click **Yes**.

Working with Attributes

The Object Type designer lets you change the following attribute details:

- **ID:** specifies the identifier used to uniquely identify the attribute.
- **Display Name:** specifies the name used to display the attribute in Orchestrator.
- **Data Type:** the type of data that the attribute represents.
- **SREL Table:** specifies the object type that this attribute references. Only available, when *SREL* is selected as the data type.
- **Default Value:** The default value to assign to the attribute when it is selected in Orchestrator.
- **Required:** indicates whether the attribute is required when creating new instance of the object type.
- **Read Only:** indicates whether the attribute can be modified.
- **Value Browser:** indicates whether the attribute should have a list browser when it is selected in Orchestrator. **Important:** Value browsers are only available for srel attributes and for performance reasons should only be enabled when the object type that is being referenced contains a relatively small number of objects.
- **Derived:** indicates that the attribute is derived from other attributes. Note that derived attributes do not support filtering.

Add a new attribute:

1. Select the object type that will contain the new attribute in the **Object Type List**.
2. Click the **Add** button below the **Attribute List**. The **Attribute Details** dialog should appear.
3. Fill in the details of your new attribute.
4. Click **OK**. The new attribute should appear in the lower Attribute List.

Change an existing attribute:

1. Select the attribute that you want to change in the **Attribute List**.
2. Click the **Edit** button below the Attribute List. The **Attribute Details** dialog should appear.
3. Make any desired changes.
4. Click **OK**.

Removing an attribute

You can remove one or more attributes from an object type, although care should be taken when doing so as not to break the foreign key (SREL) relationships between the various object types in your CA Service Desk environment.

Remove an attribute:

1. Select the attribute(s) that you want to remove and click the **Remove** button just below the **Attribute List**. Alternatively, you can press the **DEL** key.
2. When prompted, click **Yes**,