

# **Deltek Vision® 6.1**

## **Connect for Microsoft Outlook Presets Configuration Administrator Guide**

**May 17, 2010**



13880 Dulles Corner Lane  
Herndon VA 20171  
TEL: 703.734.8606  
FAX: 703.734.1146

While Deltek has attempted to verify that the information in this document is accurate and complete, some typographical or technical errors may exist. The recipient of this document is solely responsible for all decisions relating to or use of the information provided herein.

The information contained in this publication is effective as of the publication date below and is subject to change without notice.

This publication contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, or translated into another language, without the prior written consent of Deltek, Inc.

This edition published May 2010.

© 2010 Deltek, Inc.

Deltek's software is also protected by copyright law and constitutes valuable confidential and proprietary information of Deltek, Inc. and its licensors. The Deltek software, and all related documentation, is provided for use only in accordance with the terms of the license agreement. Unauthorized reproduction or distribution of the program or any portion thereof could result in severe civil or criminal penalties.

All trademarks are the property of their respective owners.

# Contents

Overview ..... 1

    What are Presets?..... 1

    Type Hierarchy and Required Types ..... 1

    Composing Restrictions from the Hierarchy ..... 3

Preset Definition ..... 4

    Preset XML Format ..... 4

    Restriction XML Format ..... 5

Fields Available for Defining Presets..... 11

    Client ..... 11

    Vendor ..... 12

    Contact ..... 14

    Opportunity ..... 15

    Activity ..... 16

    Employee ..... 16

User Independent Presets..... 18



## Overview

This guide describes how to configure Deltek Vision Connect for Microsoft Outlook presets to implement both default and custom filters for Connect for Microsoft Outlook installations.

### What are Presets?

Presets are a component of the customization package and are defined in the following file:

- **connector\_configuration.xml**

Presets allow you to switch to a pre-defined filter configuration from the Control Panel dialog, thus allowing you to avoid the complex task of building the correct filters from scratch.

During installation, or when a new customization package is applied, the very first preset defined in the package is activated by default, so you start the synchronization process with realistic filters that narrow your data set (that is, you are not synchronizing the whole Vision database).

Experienced users can customize any filter by adding, modifying, or removing criteria in the Control Panel. However, the presets themselves cannot be changed by end users. Your system administrator will modify presets and publish a customization package with new presets, as needed.

Because presets are a component of the customization package, different users/roles may receive different presets as part of the customization.

Presets can be defined independent of the target user—that is, they do not have to contain a hardcoded user ID. Instead, they can refer to **current user**, which is actualized to the particular user ID at the time the preset is applied.

The synchronization engine requires that every type, down to the child object (like Client Alias) or association, be assigned a specific restriction to retrieve objects. In most cases the restriction for children types can be derived from restrictions for parent types at the time the preset is applied. As a result, a partial preset definition is allowed for some types.

### Type Hierarchy and Required Types

The preset structure and the set of required types depend on the type hierarchy defined by the connector, which also controls how types are presented in the Control Panel, which fields are available for filtering, and so on. Any changes to your data model may require updates to the presets to support the new model.

The currently implemented type hierarchy is outlined below.



With the types in brackets hidden, this tree representation is the same as the one that appears on the Control Panel.

- **Activity**
  - [ContactActivity]
  - [EMActivity]
- **Client**
  - [CLAddress]
  - [ClientFileLinks]

- [ClientAlias]
- **Activity**
- [EMClientAssoc]
- [OpportunityClientAssoc]
- [PRClientAssoc]
- **Contacts**
- **Contacts**
  - [CLAddress]
  - [VEAddress]
  - [ContactFileLinks]
  - [EMContactsAssoc]
  - [OpportunityContactAssoc]
  - [PRContactAssoc]
  - [ContactActivity]
- **Opportunity**
  - [OpportunityClientAssoc]
  - [OpportunityContactAssoc]
  - [OpportunityEMAssoc]
  - [OpportunityFileLinks]
  - [OpportunityVEAssoc]
  - **Activity**
- **Vendor**
  - [VEAddress]
  - [VEFileLinks]
  - [VendorAlias]
  - [VEProjectAssoc]
  - [OpportunityVEAssoc]
  - **Contacts**
- **Employee**
- **Project**

Types displayed in **Bold** in this list:

- Have visual representation on the Control Panel.
- Require that a preset restriction be defined.

Types displayed in [square brackets] in this list:

- Do not have visual representation on the Control Panel.
- Do not require that a preset restriction be defined, because it can be generated from the parent type.

## Composing Restrictions from the Hierarchy

When you save changes on the Control Panel, Connect for Microsoft Outlook performs a special process to combine restrictions for the same type set on different levels into a complex restriction (one per type).

For example in the tree presented in the previous section, type **Activity** exists on the top level and also under **Clients** and **Opportunities**. As a result, the restriction for type **Activity** will contain three components combined by **OR**, as follows:

- **Final Restriction** (Activity) = **Restriction** (Activity top level) OR **Restriction** (Activity for matched Clients) OR **Restriction** (Activity for matched Opportunities).

The latter two components of this restriction are examples of Foreign Key restrictions, which are defined later in this document.

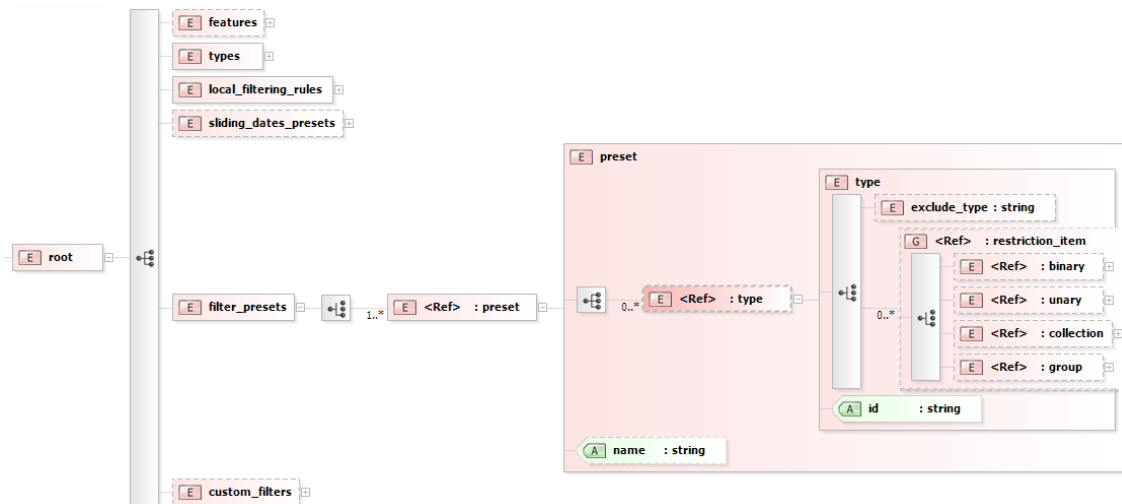
These rules should be followed when you define presets so that the Control Panel can parse presets correctly. Using a restriction processing code that generates child restrictions from parent types significantly simplifies this task, because there is no need to take any action for non-visual types. However, the hierarchical display of visual types requires a higher level of attention to ensure the correct representation of the restriction in XML.

## Preset Definition

Presets are defined manually in XML. Each XML file in the customization package has XSD schema defined, which allows the XML files to be validated against XSD.

### Preset XML Format

Presets are defined under the **<filter\_presets>** tag, which is located under the document root of the **connector\_configuration.xml** file.



- The tag **<filter\_presets>** is a container for the set of **<preset>** tags defined.
- Each **<preset>** has a name (shown to user) and a collection of **<type>** tags.
- Each **<type>** has an ID (internal) and either a sub tag **<exclude\_type>** (indicating that the type should not be checked on the Control Panel) or a definition of the restriction for that type
- If a particular type is not provided, all records of this type will be retrieved during synchronization.

### Sample XML

The following sample XML defines a preset named **Test filters** that contains two restrictions—an explicit no match for Opportunity, and a restriction for Action.

```
<root>
...
  <filter_presets>
    <preset name="Test filters">
      <type id="Opportunity">
        <exclude_type/>
      </type>
      <type id="Action">
```



```

...
    </type>
  </preset>
</filter_presets>
...
</root>

```

## Restriction XML Format

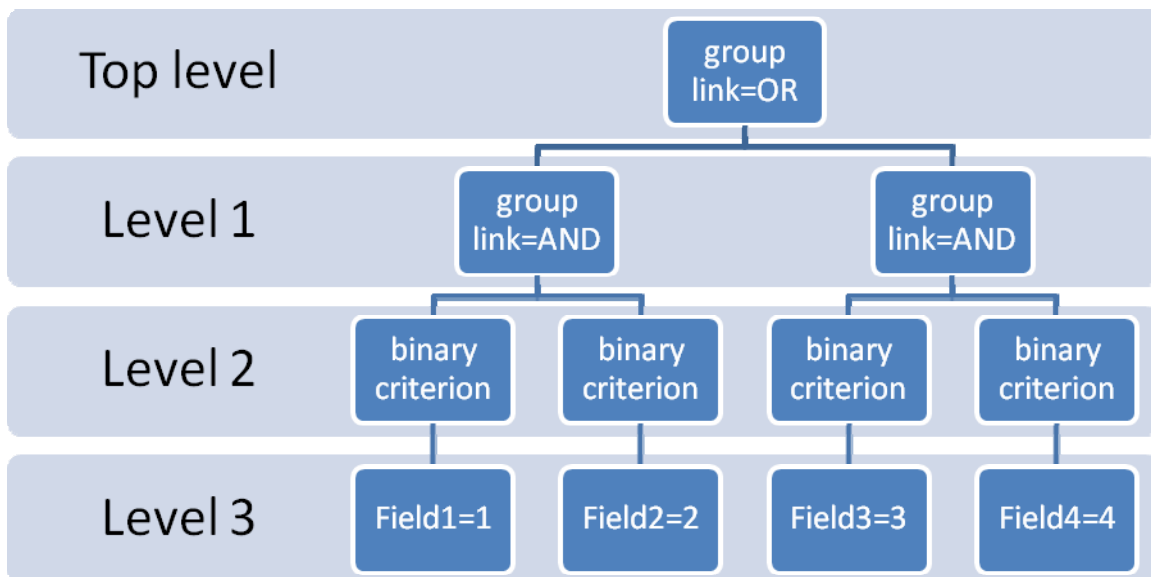
Restriction is defined as a hierarchical tree of groups and criterions. Nodes from the same level can be linked either by **AND** or by **OR**. Some nodes may have children nodes.

The following types of nodes are available:

- **Group** — Group can contain any other node including group itself
- **Binary** — Binary criterion, contains reference to field, to value and comparison operation
- **Unary** — Unary criterion, contains reference to field and comparison operation
- **Collection** — Foreign key criterion, defines linkage between two entities

This model allows representing any restriction as a tree.

For example, restriction **((Field1=1 AND Field2=2) OR (Field3=3 AND Field4=4))** would appear as follows in a tree format:

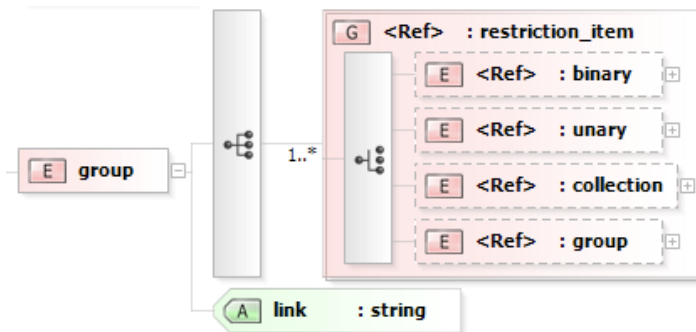


This tree structure is encoded to XML as:

```
<root>
...
  <type id="Opportunity">
    <group link="or">
      <group link="and">
        <binary field="Field1" condition="eq">
          <value type="integer">1</value>
        </binary>
        <binary field="Field2" condition="eq">
          <value type="integer">2</value>
        </binary>
      </group>
      <group link="and">
        <binary field="Field3" condition="eq">
          <value type="integer">3</value>
        </binary>
        <binary field="Field4" condition="eq">
          <value type="integer">4</value>
        </binary>
      </group>
    </group>
  </type>
...
</root>
```

## The <group> Tag and Nesting

The <group> tag allows you to define a group of nodes, linked by either **AND** or **OR**. Nested groups are possible.



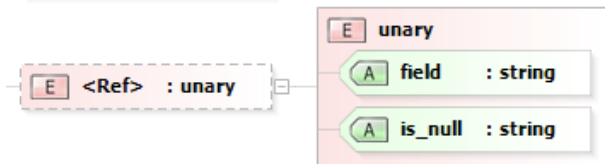
```
<group link="and">
  <group link="or">
    . . .
  </group>
</group>
```

Allowed values for **link** attribute are:

- AND
- OR

## The <unary> Criterion

The <unary> tag allows you to define unary criterion, comprised of field names and operations. For unary restriction, only **IS NULL** and **IS NOT NULL** operations are supported.



```
<group link="and">
  <unary field="Field5" is_null="true"></unary>
  <unary field="Field6" is_null="0"></unary>
</group>
```

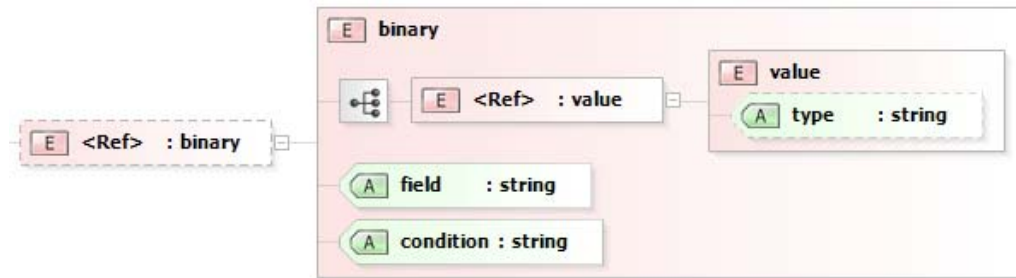
The above example can be translated as: **Field5 IS NULL AND Field6 IS NOT NULL.**

The allowed values for the **is\_null** attribute are:

- 0
- 1
- true
- false

## The <binary> Criterion

The <binary> tag allows you to define binary criterion, comprised of field names, values, and operations (conditions).



```
<binary field="Field6" condition="eq">
  <value type="integer">6</value>
</binary>
```

The following set of conditions is supported:

- lt — less than
- gt — greater than
- le — less or equal
- ge — greater or equal
- eq — equal
- ne — non-equal
- like — SQL LIKE analogue
- in — one of the listed

The value is stored in a separate **<value>** tag. Values may have one of the following types:

- string
- integer
- bool
- double
- datetime
- array
- binary
- function

Type **array** indicates that the **<value>** tag may contain children **<value>** tags describing array elements:

```
<binary field="Status" condition="in">
```

```

<value type="array">
  <value type="string">Accepted</value>
  <value type="string">Pending</value>
  <value type="string">Rejected</value>
  <value type="string">Rerouted</value>
</value>
</binary>

```

The **function** type allows special function output to be embedded as values of the binary criterion. This function is evaluated at the time the preset is applied and the result of the function is stored in the restriction. The only function available is **today** (which returns the current date).

The following snippet will be actualized so that the restriction derived from this preset will compare **Planned** against the date when the preset was applied, and only newer records will get selected.

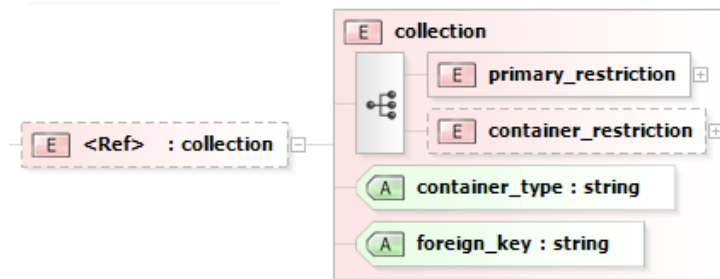
```

<binary field="Planned" condition="ge">
  <value type="function">today</value>
</binary>

```

## Foreign Key Restriction

Foreign key restriction provides a way to limit records by leaving only records which are children to selected parent types.



This is achieved by creating this restriction from two restrictions: one restriction is set for parent type (container\_restriction); another is set on the type being restricted (primary\_restriction). Both restrictions are located under the **<collection>** tag. A record matches the restriction only if the record matches the primary restriction and the specified record parent matches the container restriction.

The **<collection>** tag specifies the target parent type (**Account** in the example below), and the field holding the foreign key in a children object pointing to a parent object (**AccountId** in the example below)

```

<collection container_type="Account" foreign_key="AccountId">
  <primary_restriction>
    <group link="and">
      </group>
    </primary_restriction>

```

```
<container_restriction>
  <group link="and">
    </group>
  </container_restriction>
</collection>
```

Foreign key restriction is an effective way of representing hierarchy relationships in which only children under selected parents need to be selected. This way, the container restriction is set to limit parents as needed, and primary restriction can further limit objects being retrieved.

For hidden types, primary restriction is always set to **return all values**, which means all children of matched parents will be retrieved. For non-hidden types, you can further limit children objects, so primary restriction can be set to a specific type.

Each restriction may contain multiple Foreign Key restrictions. The container restriction of a Foreign Key restriction may also contain another Foreign Key restriction (this approach is used to support a hierarchy of types).

## Fields Available for Defining Presets

Vision API metadata defines set of fields available for defining presets. This list is modified by connector as follows:

- Some fields get hidden from the InvisibleSync data model.
- Some artificial fields may be introduced by the connector/API.
  - For example, **LinkedToMe**.
- Some fields get denormalized to respective parent objects.
  - CAddress and VAddress fields are propagated to parents (Client, Vendor, Contact).

The following tables list the fields available for filtering.

### Client

Type Name	Field Name	Filtered As Type
Client	Name	String
Client	Number	String
Client	Client Flag	Boolean
Client	Vendor Flag	Boolean
Client	Vendor	Foreign Key to Vendor
Client	Parent	Foreign Key to Vendor
Client	Type	Picklist
Client	Website	String
Client	Status	Picklist
Client	Relationship	Picklist
Client	Disadvantaged Business	Boolean
Client	Small Business	Boolean
Client	Veteran Owned Small Business	Boolean
Client	Dis. Vet. Owned Small Business	Boolean
Client	Minority Business	Boolean
Client	Woman Owned	Boolean
Client	HBCU	Boolean

Type Name	Field Name	Filtered As Type
Client	Alaska Native	Boolean
Client	Specialty	Picklist
Client	Prior Work	Boolean
Client	Recommended	Boolean
Client	Address.Description	String
Client	Address.Primary	Boolean
Client	Address.Accounting Only	Boolean
Client	Address.Billing	Boolean
Client	Address.Address1	String
Client	Address.Address2	String
Client	Address.Address3	String
Client	Address.Address4	String
Client	Address.City	String
Client	Address.State	Picklist
Client	Address.ZIP	String
Client	Address.Country	Picklist
Client	Address.Phone	String
Client	Address.Fax	String

## Vendor

Type Name	Field Name	Filtered as Type
Vendor	Name	String
Vendor	Number	Number
Vendor	Associated Client	Foreign Key to Client
Vendor	Organization	Picklist
Vendor	Website	String



Type Name	Field Name	Filtered as Type
Vendor	Status	Picklist
Vendor	Minority Status	Boolean
Vendor	Disadvantage Business	Boolean
Vendor	Small Business	Boolean
Vendor	Veteran Owned Small Business	Boolean
Vendor	Disabled Veteran Owned Small Business	Boolean
Vendor	Minority Business	Boolean
Vendor	Woman Owned	Boolean
Vendor	HBCU	Boolean
Vendor	Alaska Native	Boolean
Vendor	Specialty	Picklist
Vendor	Prior Work	Boolean
Vendor	Recommended	Boolean
Vendor	Address.Description	String
Vendor	Address.Primary	Boolean
Vendor	Address.Accounting Only	Boolean
Vendor	Address.Billing	Boolean
Vendor	Address.Address1	String
Vendor	Address.Address2	String
Vendor	Address.Address3	String
Vendor	Address.Address4	String
Vendor	Address.City	String
Vendor	Address.State	Picklist
Vendor	Address.ZIP	String
Vendor	Address.Country	Picklist

Type Name	Field Name	Filtered as Type
Vendor	Address.Phone	String
Vendor	Address.Fax	String

## Contact

Type Name	Field Name	Filtered as Type
Contact	First Name	String
Contact	Last Name	String
Contact	Middle Name	String
Contact	Preferred	String
Contact	Title	String
Contact	Type	Boolean
Contact	Client	Foreign Key to Client
Contact	Vendor	Foreign Key to Client
Contact	Email	String
Contact	Business Phone	String
Contact	Business Fax	String
Contact	Mobile	String
Contact	Home	String
Contact	Pager	String
Contact	Status	Picklist
Contact	BusinessAddress.Description	String
Contact	BusinessAddress.Address1	String
Contact	BusinessAddress.Address2	String
Contact	BusinessAddress.Address3	String
Contact	BusinessAddress.Address4	String
Contact	BusinessAddress.City	String

Type Name	Field Name	Filtered as Type
Contact	BusinessAddress.State	Picklist
Contact	BusinessAddress.ZIP	String
Contact	BusinessAddress.Country	Picklist

## Opportunity

Type Name	Field Name	Filtered As Type
Opportunity	Name	String
Opportunity	Number	String
Opportunity	Organization	Picklist
Opportunity	Primary Client	Foreign Key to Client
Opportunity	Primary Contact	Foreign Key to Contact
Opportunity	Project Manager	Foreign Key to Employee
Opportunity	Supervisor	Foreign Key to Employee
Opportunity	Principal	Foreign Key to Employee
Opportunity	Regular Project	Foreign Key to Project
Opportunity	Status	Picklist
Opportunity	Stage	Picklist
Opportunity	Type	Picklist
Opportunity	Source	Picklist
Opportunity	Date Open	Date
Opportunity	Date Closed	Date
Opportunity	Revenue	Number
Opportunity	Probability	Picklist
Opportunity	Estimated Start	Date
Opportunity	Estimated Completion	Date
Opportunity	CSI Contract	Number

Type Name	Field Name	Filtered As Type
Opportunity	Contract Vehicle	Picklist
Opportunity	Contract Type	Picklist
Opportunity	Procurement Type	Picklist
Opportunity	Proposal Due Date	Date
Opportunity	BAFO Date	Date
Opportunity	Award Date	Date

## Activity

Type Name	Field Name	Filtered As Type
Activity	CreateDate	Date
Activity	Subject	String
Activity	Location	String
Activity	Type	Picklist
Activity	Priority	Picklist
Activity	Completed	Boolean
Activity	Private	Boolean
Activity	Client	Foreign Key to Client
Activity	Primary Contact	Foreign Key to Contact
Activity	Opportunity	Foreign Key to Opportunity
Activity	Project	Foreign Key to Project
Activity	Owner	Foreign Key to Employee
Activity	StartDate	Date

## Employee

Type Name	Field Name	Filtered As Type
Employee	FirstName	String

Type Name	Field Name	Filtered As Type
Employee	LastName	String
Employee	Number	Number
Employee	Email	String
Employee	WorkPhone	String
Employee	MobilePhone	String
Employee	Status	Picklist
Employee	CreatedBy	Foreign Key to Employee

## User Independent Presets

To define a user independent preset, the restriction in the preset should refer to the current user as follows:

- **\$CurrentUserId\$**

This value is replaced with the real user ID when the preset is applied, thus allowing each user to get different restrictions based on his/her unique user ID.