WhereScape 3D

Using Business Key Collision Codes (BKCC) in WhereScape 3D

> **DISCLAIMER**
>
> The modifications suggested in this guide have worked for infoVia, but no assurances are provided. Individual WhereScape environments can differ and there is potential for these changes to conflict with other customizations, cause environmental oddities, or just not work.
> *USE AT YOUR OWN RISK*

A Business Key Collision Code, or BKCC for short, is a Data Vault technique to allow greater control of passive integration. This is not something natively available in WhereScape 3D but it can be implemented with a few modifications to the environment and model conversion rules.

Overview

This modification works by adding a new attribute type called "BKCC" to a business key. The value of an attribute with the BKCC attribute type is set by a data transformation using the assigned BKCC value of the connection. Alternately, an entity may specify a BKCC value for a source mapping set by adding an extended property. A BKCC value set in this way will take precedence over the connection's BKCC value. A BKCC acts as a compound business key and will work in the standard WhereScape Red environment with no modifications.

Before you begin:
Download the model conversion rule ws3d_rvls_infoVia - BKCC from the infoVia blog
Familiarize yourself with managing model conversion rules (see WhereScape 3D help Model Conversions>Managing Model Conversions)

User Guide

Setting an attribute to BKCC
After completing the **setup** (see section "BKCC Setup in 3D" below) there will be a new BKCC attribute type available in the Data Vault Design model. To make an attribute a BKCC, simply create an entity attribute and assign the attribute types of business key, or link business key for link entities, and BKCC. Select an appropriate data type for your BKCC standards. This attribute will have no source mapping. The BKCC should be the last business key listed in any set of business keys to ensure consistent hashing.

Customer

N	Customer Key	int	Customer.Customer Key	
N	BKCC	varchar(255)		
N	WWI Customer ID	int	Customer.WWI Customer ID	
N	Customer	nvarchar(100)	Customer.Customer	
N	Bill To Customer	nvarchar(100)	Customer.Bill To Customer	
N	Category	nvarchar(50)	Customer.Category	
N	Buying Group	nvarchar(50)	Customer.Buying Group	
N	Primary Contact	nvarchar(50)	Customer.Primary Contact	
N	Postal Code	nvarchar(10)	Customer.Postal Code	
N	Valid From	datetime2(27)	Customer.Valid From	
N	Valid To	datetime2(27)	Customer.Valid To	
N	Lineage Key	int	Customer.Lineage Key	

Customer properties

Entity | **Attributes** | Query | Data characteristics | Data transformations | Index | Relationship | Source mappings | Extended properties

Attributes

Attribute name:
- Customer Key
- BKCC
- WWI Customer ID
- Customer
- Bill To Customer
- Category
- Buying Group
- Primary Contact
- Postal Code
- Valid From
- Valid To
- Lineage Key

Buttons: Add, Rename, Copy, Delete, Move to top, Move up, Move down, Move to end

Created by: User defined
Label:
Data type: varchar
Size: 255
Scale:
Attribute type: Business key; BKCC
Rating:
Nullable: ☐
Unique: ☐
Primary key: ☐
Auto increment: ☐
Default value:
Comment:
Hidden: ☐

Subjects: General info
Document general information:

OK | Cancel | Apply

City

N	City Key	int	City.City Key
N	BKCC		
N	WWI City ID	int	City.WWI City ID
N	City	nvarchar(50)	City.City
N	State Province	nvarchar(50)	City.State Province
N	Country	nvarchar(60)	City.Country
N	Continent	nvarchar(30)	City.Continent

Order

N	City Key	int	Order.City Key
N	BKCC_City		
N	Customer Key	int	Order.Customer Key
N	BKCC_Customer		
N	Order Key	bigint identity	Order.Order Key
N	Stock Item Key	int	Order.Stock Item Key
N	Order Date Key	date	Order.Order Date Key

Customer

|			
N	Customer Key	int	Customer.Customer Key
N	BKCC	varchar(255)	
N	WWI Customer ID	int	Customer.WWI Customer ID
N	Customer	nvarchar(100)	Customer.Customer
N	Bill To Customer	nvarchar(100)	Customer.Bill To Customer
N	Category	nvarchar(50)	Customer.Category
N	Buying Group	nvarchar(50)	Customer.Buying Group

Setting a specific BKCC value

The default value of a BKCC is set by the value of the BKCC field in the connection documentation and applies to any data source mapping using that connection. It is possible to specify the BKCC value for any source mapping set by setting an extended property for an entity in the Data Vault Design or Data Vault model. The extended property should be set with the following values:

- Attribute - Left blank
- Property - Must be set to the source mapping set name. Case sensitive.
- Type - set as "BKCC". *Case sensitive.*
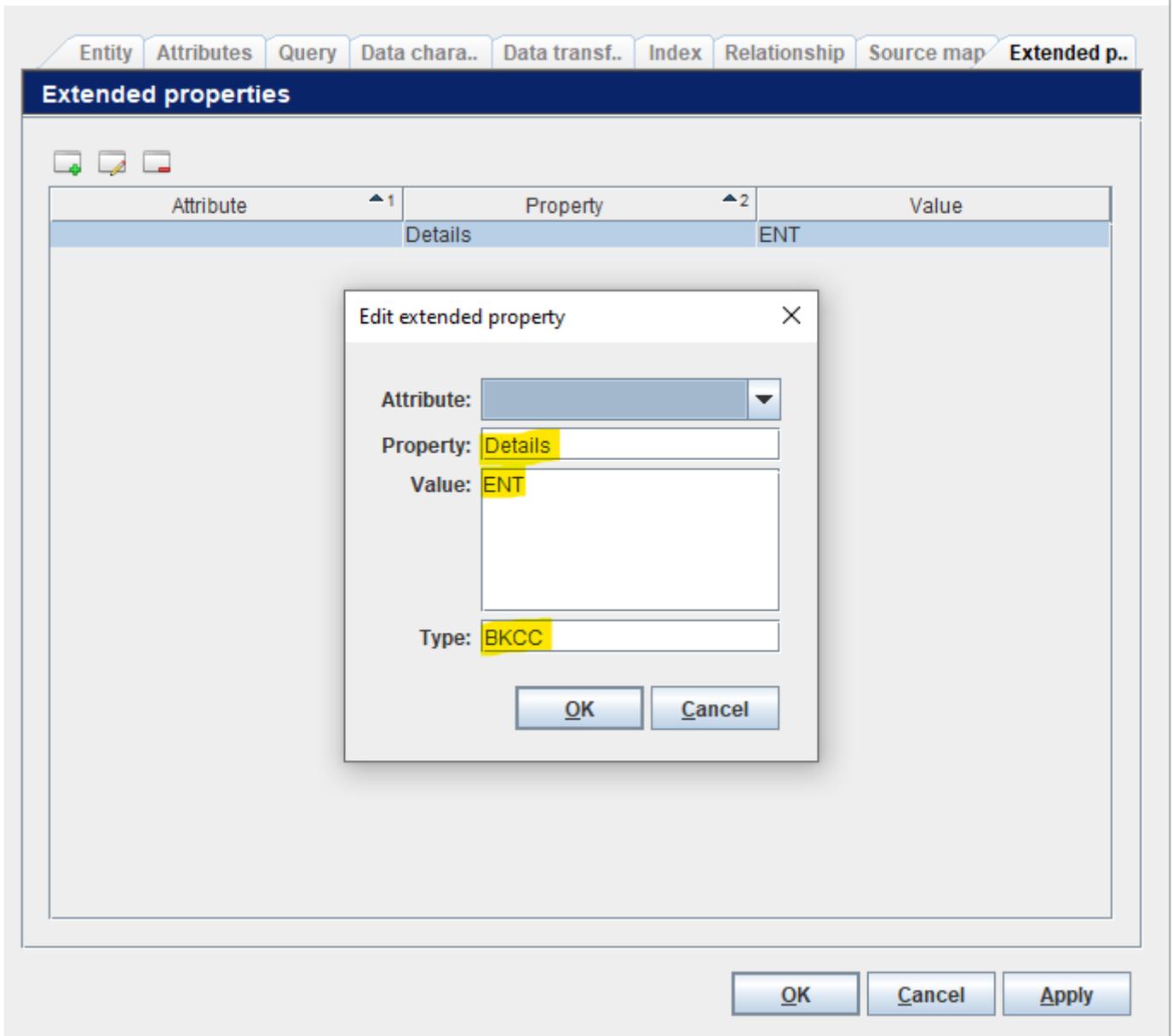
During the Load and Stage model conversion a data tranformation will be created for all BKCC attribute types based on the extended property BKCC value or, if the extended property is not set, the connection BKCC value. To view the transformation open a stage table entity's properties and select the Data Tranformations tab as seen below.
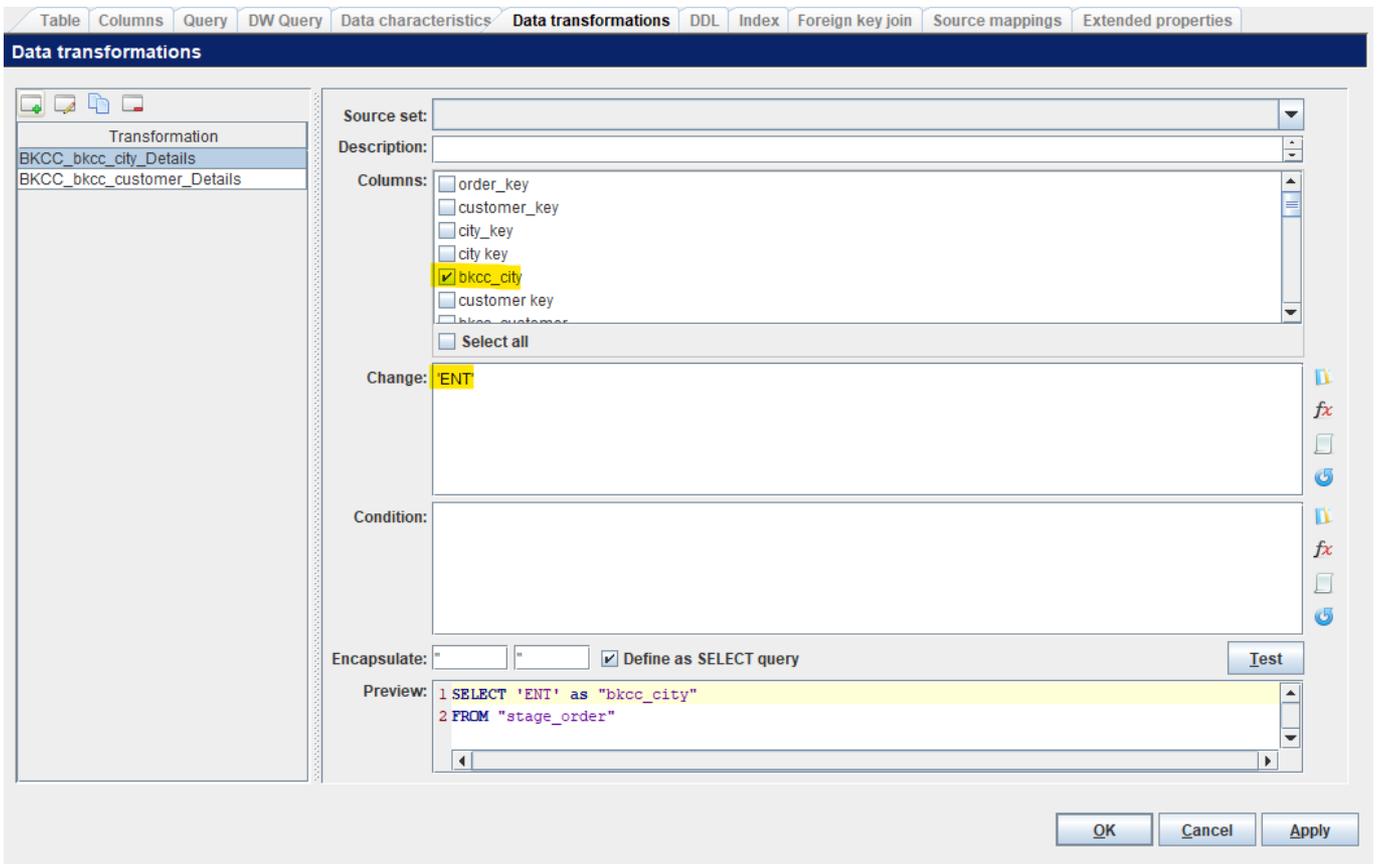
Warning: Implied link tables in Data Vault Design

In the data vault design model it is possible to create an entity that has an attribute with the type business key and an attribute with the type link business key. This will result in a link being automatically generated during the data vault model conversion process. In this situation both the BKCC any extended properties will need to be set in the Data Vault Model. Explicity forming link entities in the data vault design model ensures this will not occur.

Example: On the left is the data vault design entity and the resulting link after the data vault model conversion. Note that bkcc_order loses the BKCC attribute type in the process.

BKCC Setup in 3D

Import

Import the Model Conversion Rule (MCR) file into your WhereScape 3D Environment

Select Tools > Manage Model Conversions

Use the green arrow to import the downloaded Model Conversion Rule "ws3d_rvls_infoVia - BKCC"
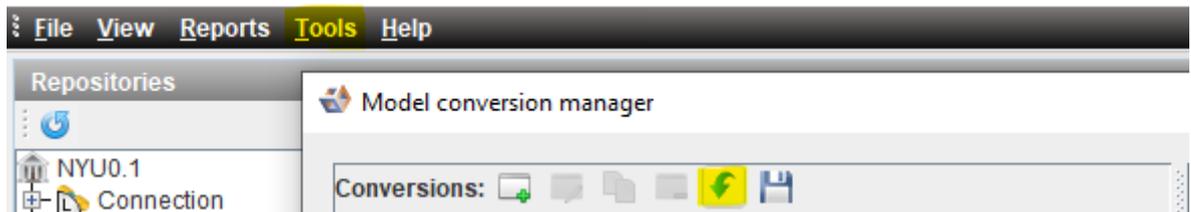
Environment Settings

Add user documentation options
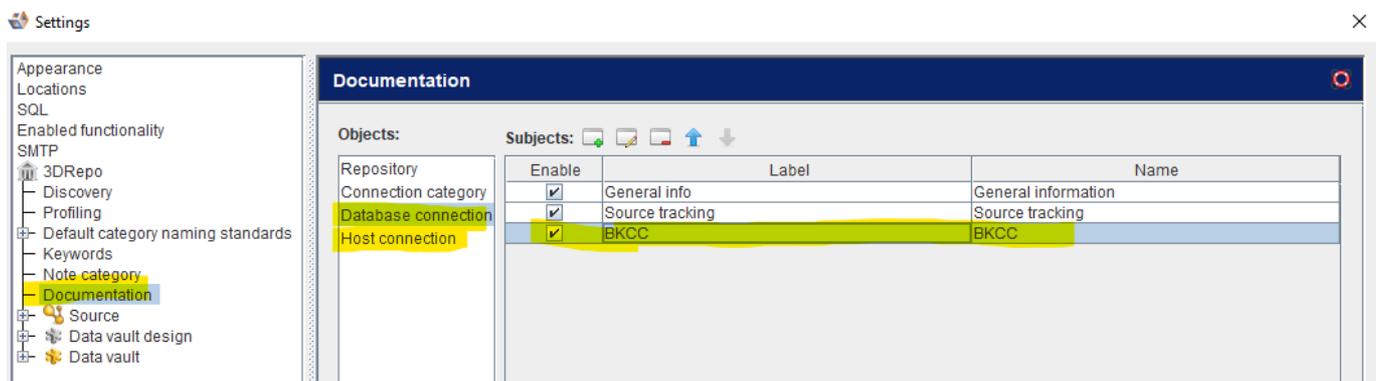
Open settings (Tools>Settings)

Navigate to the "Documentation" section in the left window

Select "Database connection" in the Objects list.

Select the window with a green plus and enter the label "BKCC" with the description "BKCC". This is case sensitive.

Repeat for "Host connection" if needed.

Click OK

Alter "User Documentation" for connections

Right click on a connection (under repositories) and select "Enter User Documentation"

Navigate to the "BKCC" tab and enter the desired default Business Key Collision Code for that connection.

Click OK

Repeat for other connections

Add BKCC Attribute Type

Select a Data Vault Design Model version

Select the Manage attribute types wizard from the workflow pane

Click the window with the green plus and enter "BKCC" for the name and "Business Key Collision Code" for the description.

Choose a color.

Click Modify

Add the conversion rule for Data Vault to Load and Stage

Add the model conversion rule *ws3d_rvls_infoVia - BKCC* to the model conversion rules between *ws3d_rvls - Merge and Clean up stages* and *ws3d_rvls - Create loads*. This can be done during the conversion process from a Data Vault model to Load and Stage via the "Create Load and Stage" wizard during the "Apply model conversion" portion of the wizard, as shown below

Apply model conversion

Select a model conversion

Model conversion: 🗎₊

ws3d_rvls - Create initial stages (Template) ▼ ✕

ws3d_rvls - Define change hashes on satellite stages (Template) ▼ ✕

ws3d_rvls - Define extended properties for hub and link hash key (Template) ▼ ✕

ws3d_rvls - Merge and clean up stages (Template) ▼ ✕

ws3d_rvls_infoVia - BKCC (Template) ▼ ✕

ws3d_rvls - Create loads (Template) ▼ ✕

ws3d_rvls - Housekeeping (Template) ▼ ✕

☐ Create snapshots

< Back   Next >   Cancel

Alter the model conversion rule "ws3d_grv - Logical to DV Spine"

Open Tools>Manage Model Conversions. Make a copy of *ws3d_grv - logical to DV Spine* model conversion ruleset and then adjust the created copy by replacing the code for each rule's listed field.

Rule: Delete source entity extended properties

- Add matching condition - Custom Entity Template
- Field - Entity Validation Expression

```
{%- set BKCCprop = "true" %}
{%- for srcSet in table.sourceSets -%}
    {%- set srcName = srcSet.name -%}
    {%- from table.extendedProperties as exProp where exProp.type == "BKCC" -%}
        {%- set BKCCprop = "false" -%}
    {%- endfrom -%}
{%- endfor -%}
{{BKCCprop}}
```

Create Rule: Clear Link BKCC key off Hub

Rule type - Assign Attribute Type

Position - Below "Set Link entity label to current name" and above "Clear Link business key off Hub Business key"

- Add matching condition - Entity Type
  Select type "Hub"
- Add matching condition - Attribute Type
  Select type "BKCC"
- Field - Assign
  Select BKCC and Business Key
- Field - Remove other attribute types
  Select

Rule: Clear Link business key off Hub Business key

- Add matching condition - Logical Group
  Select "IS NOT" Add matching condition - Attribute Type
  Select type "BKCC"

Create Rule: Set BKCCs in Link to Business key and BKCC

Rule type - Assign Attribute

Position - Below "Define entities as Links" and above "Set FKs in Link to Business key"

- Add matching condition - Entity Type
  Select type "Link"
- Add matching condition - Attribute Property
  Select "Foreign Key"
- Add matching condition - Attribute Type
  Select type "BKCC"
- Field - Assign
  Select BKCC and Business Key
- Field - Remove matched attribute types
  Select
- Field - Remove other attribute types
  Select

Rule: Set FKs in Link to Business key

- Add matching condition - Logical Group
  Select "IS NOT" Add matching condition - Attribute Type
  Select type "BKCC"

Final Notes:

The goal was to create an easy-to-use BKCC solution and while we have had excellent feedback from clients. If you MUST use implied links in your data vault design it is possible to add some logic in the model conversion rules to recognize BKCCs by attribute name. Feel free to contact us on the WhereScape forums (or through infoVia) if you need assistance with that.

ChangeLog: 1.1.1

- Add clarification for prerequisites
- Terminology fixes 1.1
- Removed Red version and RDBMS from document notes as neither is altered by this solution