TIMEXTENDER

HARVEST

CUSTOM DATA SOURCE PROVIDER FOR TX DWA

Contents

Introduction

The Harvest provider enables you to extract data from Harvest for use in your project through the custom data source feature of TX DWA.

The provider uses an API Harvest has a made available for extraction of data. Since Harvest preserves the right to change and update the API, the provider enables you to change the table schema to adapt to changes in the API.

The amount of calls a given application can make to through the API in a given amount of time is limited. At the time of writing, this limit is 100 calls every 15 seconds. This can of cause increase the time it takes to load data from Harvest.

Learn more about the API at the Harvest website:
http://help.getharvest.com/api/introduction/overview/introduction/

Installation

The provider is installed through the Custom Components setup application available at the TimeXtender support site: https://support.timextender.com/hc/en-us/articles/209604866

Usage

The first time you want to extract from Harvest, you need to consider authentication. Harvest offers basic username/password authentication and OAuth2. The latter is recommended by Harvest and in addition to that, the data transfer speed when using OAuth2 is estimated in tests to be twice as fast as basic security. However, you will need to set up OAuth2 before you can use it.

When you have taken care of authentication, you can then add a custom data source in TX DWA that uses the provider.

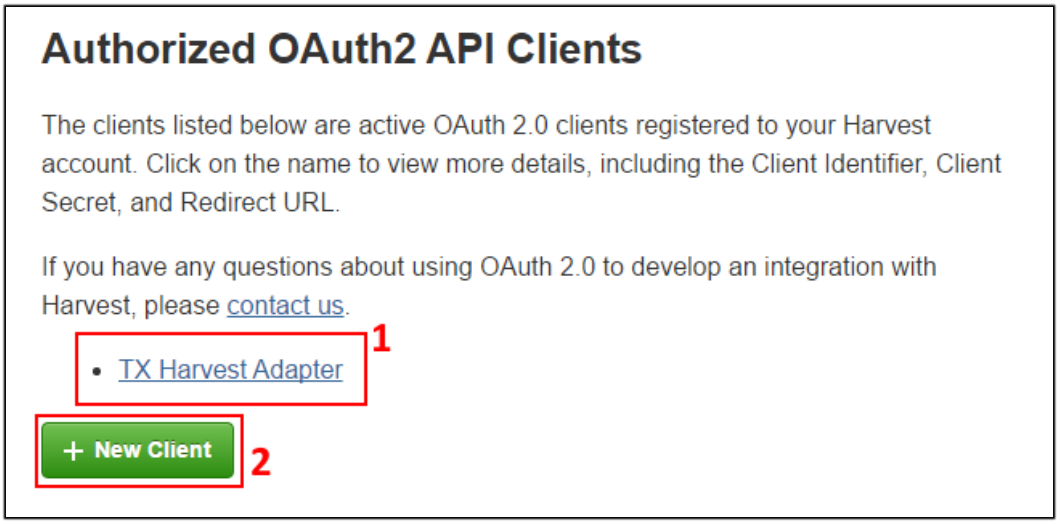
Authenticating with Harvest through OAuth2

If you want to use basic authentication, you can skip this section and proceed to Adding a Harvest Custom Data Source. To use OAuth2 for authentication, follow the steps below to register an application and get a client ID and a client secret.

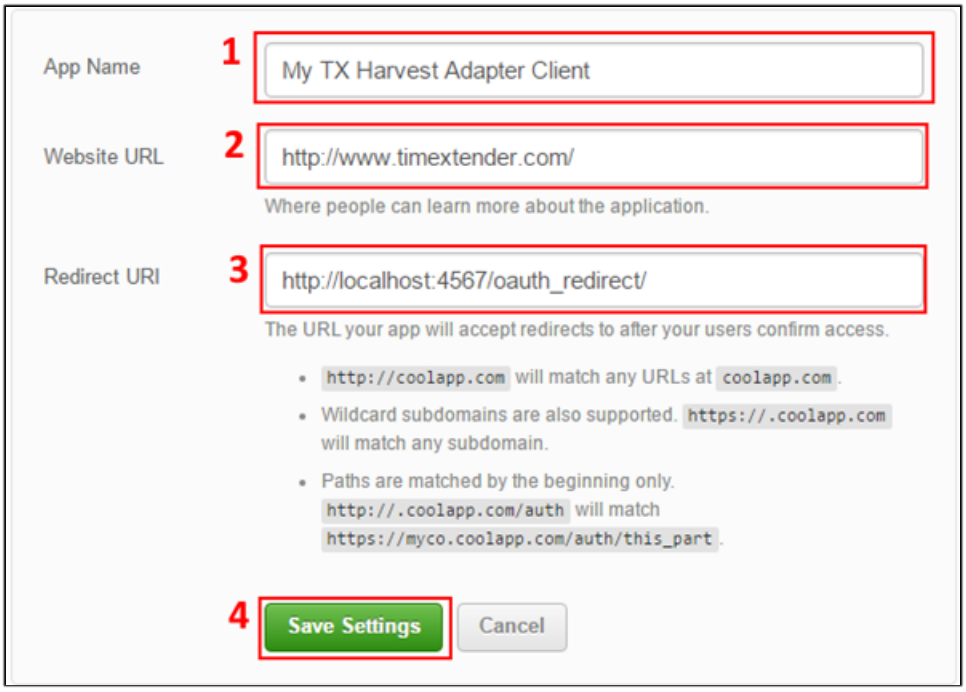
Register an API Client

To use OAuth2 as authentication you need to register an API client.

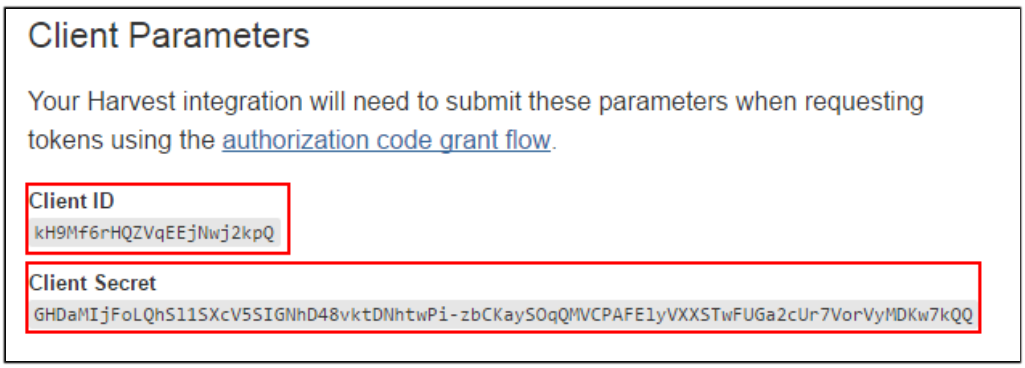
1. Go to the **Authorized OAuth2 API Clients** page in Harvest: https://platform.harvestapp.com/oauth2_clients



   The page shows a list of already existing API Clients (1). In the following, we assume that you wish to create a new client (2) and not reuse an existing client.
2. Click on **New Client**.



1. Enter a **App Name**. This name is for display purposes only.
2. Enter a **Website URL**, for instance the URL to your company website. The URL is required for registration, but not used by the provider.
3. In **Redirect URI**, enter the URL where the provider listens for authentication requests. The default and recommended value is: "http://localhost:4567/oauth_redirect/".
4. Click on **Save Settings** to create the client.
5. Scroll down to **Client Parameters** and see **the Client ID** and **Client Secret** created for the new client. These are the connection settings needed by the Harvest Provider.



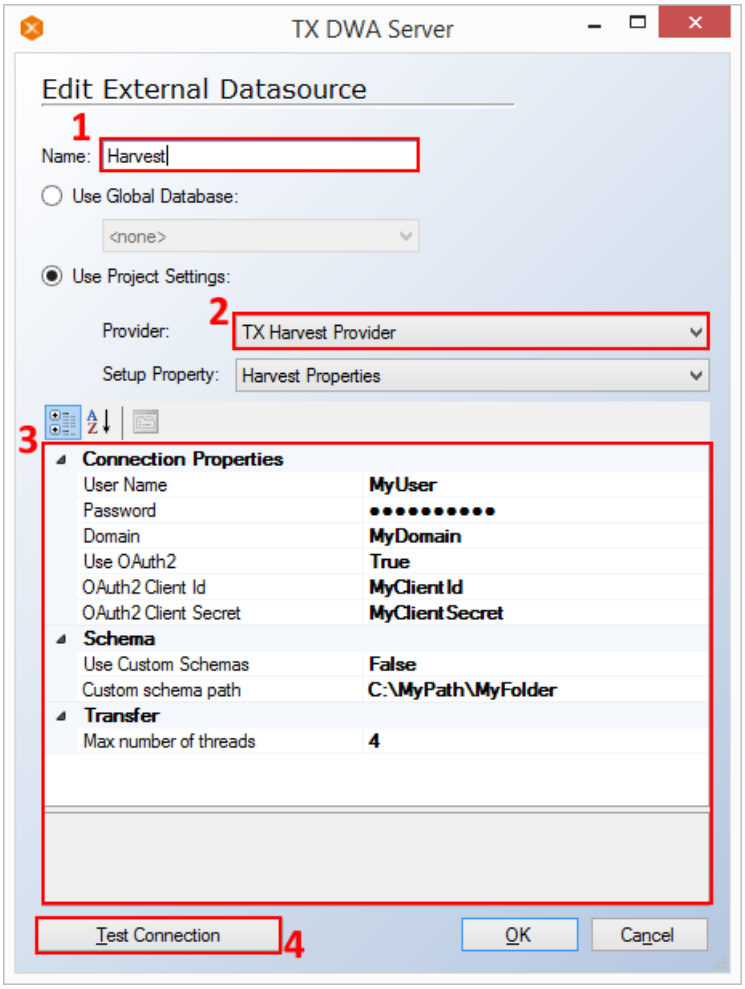
Adding a Custom Data Source

When you have created your Harvest app and have a client ID and client secret, you are ready to set up the custom data source in TX DWA.

Adding a Harvest Custom Data Source

To add and configure a custom data source, follow the steps below.

1. On the **Data** tab, right click on **Data Sources,** click on Data Sources and click on **Add Custom Data Source**. **Add Custom Data Source** opens.

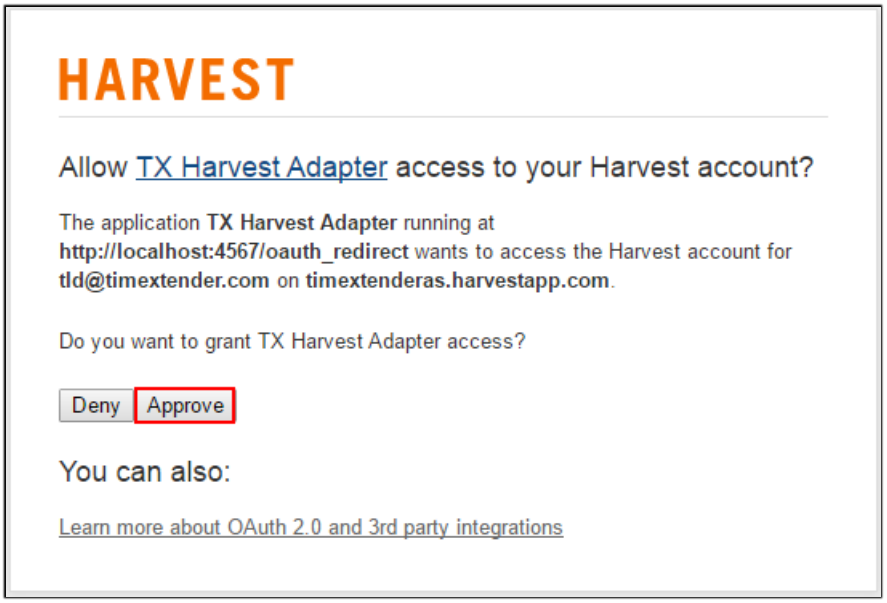


2. Type a **Name** for the data source.
3. In the provider list, click on **TX Harvest Provider**.
4. Setup the Harvest properties.
    - Connection Properties.
      If you wish to use OAuth2 for authentication, set **Use OAuth2** to **True**. Enter your **Oauth2 Client** and **OAuth2 Client Secret**.
      - OR -
      If you wish to use basic authentication, enter **User Name**, **Password** and

Domain (Your domain is the first part of your login URL, i.e. https://<domain>.harvestapp.com)

- (Optional) Schema. Set **Use Custom Schemas** to **True** to enable the use of custom created schemas and enter the path to the custom schemas in **Custom Schema Path**. Enabling this will ignore all default schemas. See Custom Schemas below for more information.
- (Optional) Transfer. Enter the maximum number of threads you wish to provider to use in the **Max Number of Threads** box. The maximum number of threads in each table for transferring data. This setting can be individually overridden for each table with the "Override Transfer Settings" command on the data source tables.

5. Click **Test Connection**. If you have OAuth2 enabled, you default web browser will open on the Harvest website. If you are not signed into Harvest already, you will be asked to do so. When you have signed in, click on **Approve** to allow the client to access your Harvest account.



This is only necessary the first time you connect through OAuth2. Please note that the authentication process has a 60 second timeout and closing the browser window without clicking **Approve** or **Deny** will result in TX DWA waiting for the timeout.

6. Click **OK** to add the custom data source.

Re-authenticating with Harvest

If you use oAuth2 for authentication and get an execution error that has to do with authentication, you can re-authenticate by right clicking the Harvest custom data source and clicking **Authorize with OAuth2**.

Selection Rules and Incremental Load

The API supports a limited number of selection rules that are evaluated execution time. It is important to note that the provider only supports one set of selection rules for each table.

You can have multiple selection rule statements in the same selection rule. The selection rules are translated from filter parameters from the API commands and the provider will override existing parameters used more than once. Supported selection rules are described in this chapter.

updated_since

The parameter "updated_since" is of the type "utc datetime" and selects all records with a value greater than the value provided by the selection rule. This makes this selection rule ideal for incremental load. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Greater" or is a source based incremental selection rule.
2) The value provided can be translated into a datetime.

The selection rule can be used on the following tables and columns in the default schema:

Table	Column
Client_Contacts	updated-at
Clients	updated-at
Expense_Categories	updated-at
Expense_Categories	updated-at
Expenses_People	updated-at
Expenses_Project	updated-at
Invoices	updated-at
People	updated-at
Projects	updated-at
Task_Assignments	updated-at
Tasks	updated-at
Timeframes_People	updated-at
Timeframes_Project	updated-at
User_Assignments	updated-at

from

The parameter "from" is of the type "date" and selects all records with a value greater or equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Greater or Equal".
2) The value provided in the selection rule can be translated into a date.

The selection rule can be used on the following tables in the default schema:

Table	Column
Expenses_People	spent-at
Expenses_Project	spent-at
Invoices	issued-at
Timeframes_People	spent-at
Timeframes_Project	spent-at

to

The parameter "to" is of the type "date" and selects all records with a value greater or equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Less or Equal".
2) The value provided in the selection rule can be translated into a date.

The selection rule can be used on the following tables in the default schema:

Table	Column
Expenses_People	spent-at
Expenses_Project	spent-at
Invoices	issued-at
Timeframes_People	spent-at
Timeframes_Project	spent-at

client

The parameter "client" is of the type "int" and selects all records with a value equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Equal".
2) The value provided in the selection rule can be translated into an int.

The selection rule can be used on the following tables in the default schema:

Table	Column
Invoices	client-id
Projects	client-id

Please note that the parameter provides data for all client ids if the client id was not found in the collection. This is an implementation in the Harvest API command set and cannot be avoided.

user_id

The parameter "user_id" is of the type "int" and selects all records with a value equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Equal".
2) The value provided in the selection rule can be translated into an int.

The selection rule can be used on the following tables in the default schema:

Table	Column
Timeframes_Project	user-id

Please note that the parameter provides data for all user ids if the user id was not found in the collection. This is an implementation in the Harvest API command set and cannot be avoided.

status

The parameter "status" is of the type "text" and selects all records with a value equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria's:

1) The selection rule is of the type "Equal".
2) The value provided in the selection rule is equal to one of these values:
    - Open
    - Partial
    - Draft
    - Paid
    - Unpaid
    - Pastdue
    - Closed

The selection rule can be used on the following tables in the default schema:

Table	Column
Invoices	state

is_closed

The parameter "is_closed" is of the type "bool/bit" and selects all records with a value equal to the value provided by the selection rule. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Equal".
2) The value provided be the selection rule is equal to one of these values:
    - True
    - Yes
    - 1
    - False
    - No
    - 0

The selection rule can be used on the following tables in the default schema:

Table	Column
Timeframes_People	is-closed
Timeframes_Project	is-closed

billable

The parameter "billable" is of the type "bool/bit" and selects all records with a value equal to the value provided by the selection rule. The selection rule is used on a column which supports multiple parameters types, so it needs an identifier in the value to select which parameter is used. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Equal".
2) The value starts with the identifier string: [billable]
3) The value provided be the selection rule proceeding the identifier is equal to one of these values:
    - True
    - Yes
    - 1
    - False
    - No
    - 0

The selection rule can be used on the following tables in the default schema:

Table	Column
Timeframes_People	is-billed
Timeframes_Project	is-billed

only_billed

The parameter "only_billed" is of the type "bool/bit (positive)" and selects all records with a value equal to the value provided by the selection rule. The selection rule is used on a column which supports multiple parameters types, so it needs an identifier in the value to select which parameter is used. The parameter is used as a selection rule with the following criteria:

4) The selection rule is of the type "Equal".
1) The value starts with the identifier string: [only_billed]
2) The value provided in the selection rule proceeding the identifier is equal to one of these values:
    - True
    - Yes
    - 1

The selection rule can be used on the following tables in the default schema:

Table	Column
Timeframes_People	is-billed
Timeframes_Project	is-billed

only_unbilled

The parameter "only_unbilled" is of the type "bool/bit (positive)" and selects all records with a value equal to the value provided by the selection rule. The selection rule is used on a column which supports multiple parameters types, so it needs an identifier in the value to select which parameter is used. The parameter is used as a selection rule with the following criteria:

1) The selection rule is of the type "Equal".
2) The value starts with the identifier string: [only_unbilled]
3) The value provided in the selection rule proceeding the identifier is equal to one of these values:
    - True
    - Yes
    - 1

The selection rule can be used on the following tables in the default schema:

Table	Column
Timeframes_People	is-billed
Timeframes_Project	is-billed

Custom Schemas

The provider supports the use of custom created schemas.

You can extract the default schemas to a folder by clicking on **Extract Default Schemas** on the data source.

The schema must be created as an XML file with the file extension ".xml" and follow the structure defined in this chapter. See an example at the end of this document.

Root Node

The root node of the schema must be named "HarvestSchema" and include the following nodes:

Node name	Node value	Optional
Table	See Table Node.	False
Columns	See Columns Node.	False
RequestIterator	See RequestIterator .	True
SchemaDependency	See SchemaDependency .	True
CustomParameters	See CustomParameters .	True

Table Node

This node describes the basic properties of the table.

The node must be named "Table" and include the following nodes:

Node name	Node value	Optional
Schema	Name of database schema.	True
Name	Name of table in database.	False
DisplayName	Name of table in TX DWA.	True
HttpRequest	Value of http request to receive data. The value is the last part of the default request (http://<domain>.harvestapp.com) including the starting slash, eg. "/Invoices"	False

Columns Node

This node is a list of all columns in the table.

The node must be named "Columns" and include a list of column nodes. Each column node must be named "Column" and include the following nodes:

Node name	Node value	Optional
Name	The name of the column in the database.	False
DisplayName	The name of the column in TX DWA.	True
SqlDataType	The SQL data type to be used in the database.	False
SourceDataType	The data type received from the http request.	True
ParameterOptions	The options for parameter usage. The parameters are used for selection rules in TX DWA. See ParameterOptions Node.	True

ParameterOptions Node

This node describes the parameters which can be used as selection rules for the column.

The node must be named "ParameterOptions" and include a list of parameter option nodes. Each column node must be named "ParameterOption" and include the following nodes:

Node name	Node value	Optional
ParameterName	The name of the parameter used in the harvest API command.	False
SupportedSelectionRuleType	The type of supported selection rule. See Supported Selection Rules.	False

Supported Selection Rules

- Equal,
- GreaterOrEqual
- LessOrEqual
- Greater

RequestIterator Node

This node describes the process of iterating through calls to receive all data for the table. This is currently necessary for the "Invoices" table since each request is limited to 50 records.

The node must be named "RequestIterator" and include the following nodes:

Node name	Node value	Optional
ParameterName	The name of the parameter used in the harvest API command.	False
StartValue	The value for the iteration. The value must be an integer and is incremented by one for each request.	False
ResultsEachRequest	The number of records received for each request. The current default value is 50 and cannot be changed.	False

SchemaDependency Node

This node describes the dependency calls for the table. Some of the harvest API table requires parameters such as Ids to extract data. For instance, Invoice_Messages is dependent on the Invoices Ids.

The node must be named "SchemaDependency" and include the following nodes:

Node name	Node value	Optional
HttpRequest	Value of http request to receive data. The value is the last part of the default request (http://<domain>.harvestapp.com) including the starting slash, eg. "/Invoices"	False
DependencyParameters	List of dependency parameter nodes. See DependencyParameter .	False
CustomParameters	See CustomParameters .	True
RequestIterator	See RequestIterator .	True

DependencyParameter Node
This node describes the parameters used for the dependency calls.

The node must be named "DependencyParameter" and include the following nodes:

Node name	Node value	Optional
DependencyTableColumnName	The name of the column of the table received in the dependency request, to be used as value of the parameter. For instance, if your table is dependent on the table Invoices, the column name would be Id, which is the Id column on the Invoice table. Each id value received is used as parameter value for the dependency requests.	False
ParameterName	The parameter name used for the request. This parameter must start and end with brackets, eg. "{invoice_id}" and must be included in the http request on the table node. See Table Node	False

CustomParameters Node

This node describes the parameters to use for each transfer request and is the equivalent to a set of static selection rules. The parameter will be overridden if the user uses selection rules in TX DWA with the parameter names corresponding to the custom parameters.

The node must be named "CustomParameters" and include the following nodes:

Node name	Node value	Optional
ParameterNames	A list of name nodes. Each name node must be named "Name" and contain the value of the parameter name used in the harvest API command.	False
ParameterValueSets	A list of value set nodes to use for each request. Each value set node must be name "ValueSet" and contain a list of value nodes named "Value". The value set nodes must contain the same amount of value nodes as the number of name nodes in the parameter names node. The value node contains the value used for the parameter in the harvest API command.	False

Custom Schema Example

```
<HarvestSchema>
  <Table>
    <Schema>my_db_schema_name</Schema>
    <Name>actual_table_name</Name>
    <DisplayName>Table_Display_name</DisplayName>
    <HttpRequest>/MyRequest/{Dependency_id_1}/{Dependency_id_2}</HttpRequest>
  </Table>
  <RequestIterator>
    <ParameterName>page</ParameterName>
    <StartValue>1</StartValue>
    <ResultsEachRequest>50</ResultsEachRequest>
  </RequestIterator>
  <SchemaDependency>
    <HttpRequest>/MyDependencyRequest</HttpRequest>
    <DependencyParameters>
      <DependencyParameter>
        <DependencyTableColumnName>DependenyTable1_ColumnValueName</DependencyTableColumnName>
        <ParameterName>{Dependency_id_1}</ParameterName>
      </DependencyParameter>
      <DependencyParameter>
        <DependencyTableColumnName>DependenyTable2_ColumnValueName</DependencyTableColumnName>
        <ParameterName>{Dependency_id_2}</ParameterName>
      </DependencyParameter>
    </DependencyParameters>
    <CustomParameters>
      <ParameterNames>
        <Name>from</Name>
        <Name>to</Name>
      </ParameterNames>
      <ParameterValueSets>
        <ValueSet>
          <Value>2010101</Value>
          <Value>2011231</Value>
        </ValueSet>
        <ValueSet>
          <Value>20150101</Value>
          <Value>99991231</Value>
        </ValueSet>
    </ParameterValueSets>
    </CustomParameters>
  </SchemaDependency>
  <CustomParameters>
    <ParameterNames>
      <Name>from</Name>
      <Name>to</Name>
    </ParameterNames>
    <ParameterValueSets>
      <ValueSet>
        <Value>00000101</Value>
        <Value>99991231</Value>
      </ValueSet>
    </ParameterValueSets>
  </CustomParameters>
  <Columns>
    <Column>
      <Name>id</Name>
      <DisplayName>Id</DisplayName>
      <SqlDataType>int</SqlDataType>
      <SourceDataType>integer</SourceDataType>
    </Column>
    <Column>
      <Name>created-at</Name>
      <SqlDataType>datetimeoffset(7)</SqlDataType>
      <SourceDataType>datetimeutc</SourceDataType>
    </Column>
    <Column>
      <Name>updated-at</Name>
      <SqlDataType>datetimeoffset(7)</SqlDataType>
```

```
      <SourceDataType>datetimeutc</SourceDataType>
      <ParameterOptions>
        <ParameterOption>
          <ParameterName>updated_since</ParameterName>
          <SupportedSelectionRuleType>Greater</SupportedSelectionRuleType>
        </ParameterOption>
      </ParameterOptions>
    </Column>
    <Column>
      <Name>spent-at</Name>
      <SqlDataType>date</SqlDataType>
      <SourceDataType>date</SourceDataType>
      <ParameterOptions>
        <ParameterOption>
          <ParameterName>from</ParameterName>
          <SupportedSelectionRuleType>GreaterOrEqual</SupportedSelectionRuleType>
        </ParameterOption>
        <ParameterOption>
          <ParameterName>to</ParameterName>
          <SupportedSelectionRuleType>LessOrEqual</SupportedSelectionRuleType>
        </ParameterOption>
      </ParameterOptions>
    </Column>
  </Columns>
</HarvestSchema>
```