Guide to Generic Table Interface

Using the Generic Table Interface (GTI) within QualiWare Lifecycle Manager to import and export data from and to QualiWare.

Contents

Figure list

Change log

Date	Change
April 2022	Updated document based upon an old version

1 About This Document

This document describes how to use the Generic Table Interface in QLM (QualiWare Lifecycle Manager).

2 Generic Table Interface

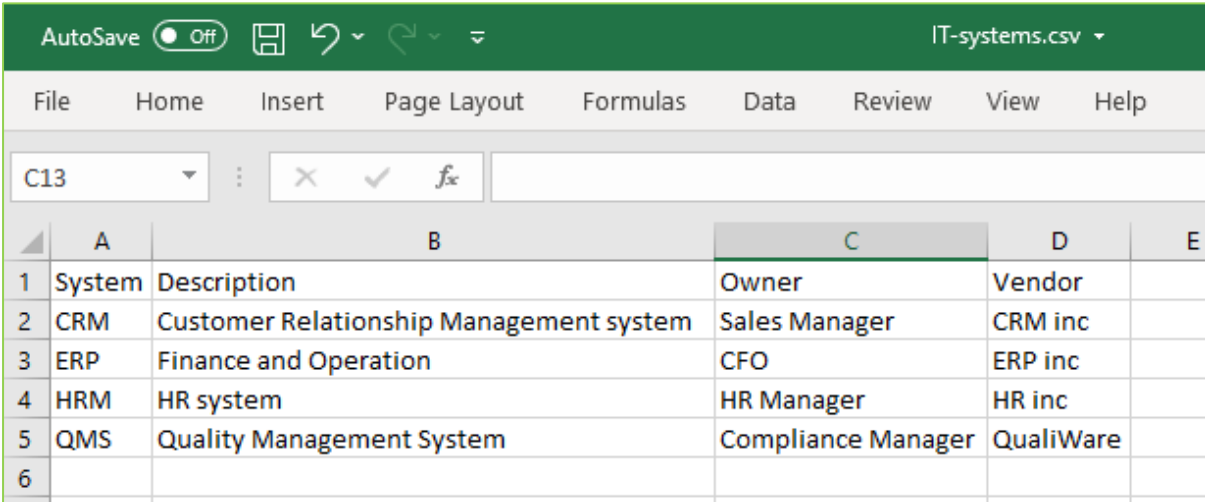The Generic Table Interface enables import or export data to and from QLM.

When you start using QLM for the first time, or when you want to expand the use of QLM, you will often have a lot of the information stored in other systems or format already, for instance in a spreadsheet or database. A quick way to initially load such data into QLM is by creating a GenericTableInterface object that can "translate" the data from a CVS-file into objects, properties and relationship between objects in QLM.

3 Preparing Files for Import to QLM using the GTI

It is possible to import data from multiple sources as instances of templates within QLM.

Data from other systems, can usually be exported to delimited text files such as a comma-separated values (CVS) file. This is the type of file can be used as input to the Generic Table Interface (GTI) within QLM.

In Figure 1 is an example of a short list of IT systems with some additional information.

Figure 1: Example of CVS-file in excel

In this example:

- the first column (i.e. *System*) represents the name of the IT-system,
- the second column (*Description*) represents a short description of the system,
- the third column (*Owner*) represents the position in the organisation that owns the system,
- and the fourth column (*Vendor*) represents the name of the system vendor.

This guide will show how these data can be imported to objects and data in QLM.

4 Utilizing the Generic Table Interface in QLM

The Generic Table Interface (GTI) in QLM is a two-way interface, in that data can be imported to QLM, or exported from QLM in a csv format.

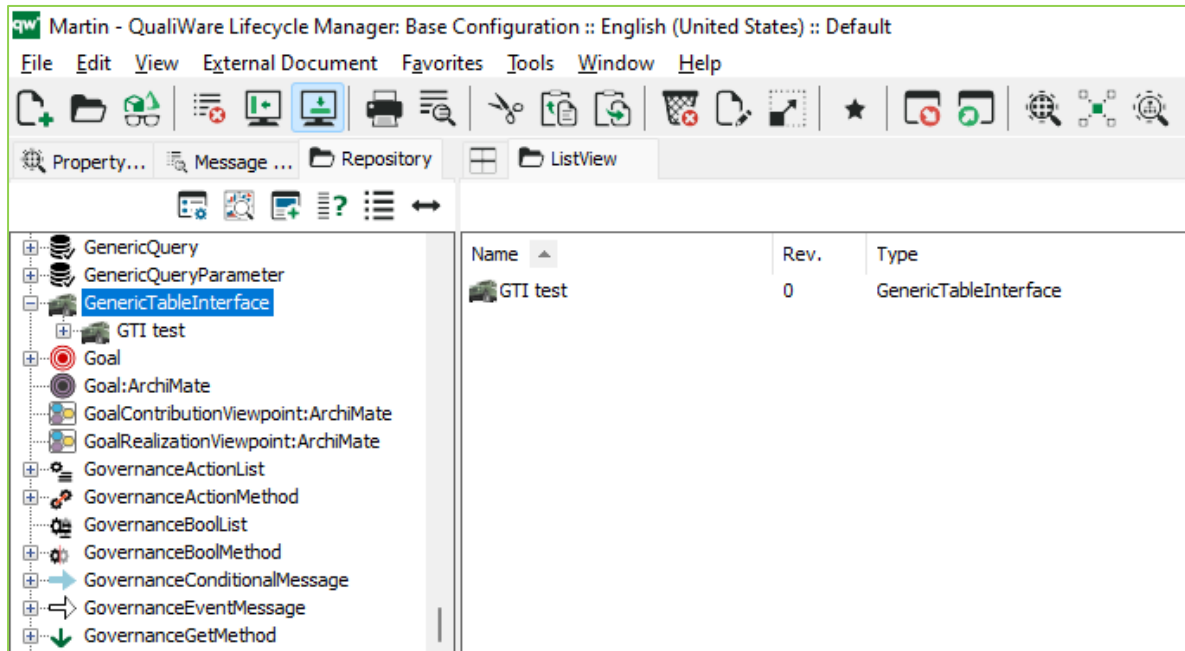The GenericTableInterface template is available in the RepositoryExplorer of QLM. See Figure 1.

Figure 2: GenericTableInterface is a template in QLM

The user selects the template from the RepositoryExplorer and builds an instance of a GTI in the repository, in the figure above the instance is named "GTI test". It is possible to have multiple GTI's in the repository and each is stored in the repository for reuse in importing data from multiple sources where that data is of a common type and structure in terms of the columns in the spreadsheet.

4.1 Import data using GTI

In this section we will describe how to use the GTI to import content from the csv file introduced in section 3.

The import is specified on the input-tab of the GTI, using the following steps, cf. Figure 3:

1. Select the **Browse button** at the top of the Input tab. This selection will permit the user to look for and select a csv file from a location on the computer or network. The field will populate after choosing the appropriate file.
2. If the file is selected via the Browse button, there is no need to check the box for "**Prompt for File**". If the Input File field is left blank and the Prompt for File checkbox is checked, then when running the GTI the user will simply be presented with a dialog to choose the input file.
3. If in Excel or Access, and the user has set the delimiter for the csv file, then the **Delimiter** choice box must be selected accordingly. Typically, this is a comma (,) or semicolon (;) and the user can choose the right one from the drop-down list.
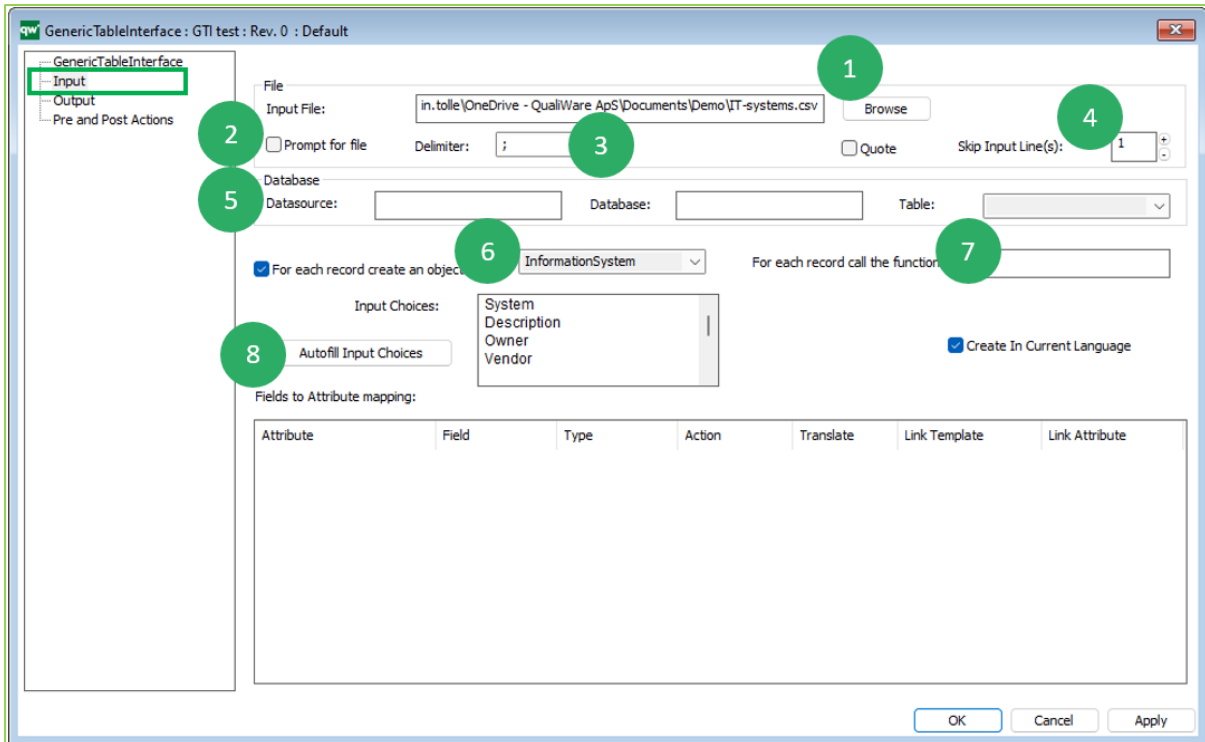
Figure 3: How to set up an input for a GTI

4. In the csv file the user must have a row where the column headings are identified, this is typically in row 1. The Skip Input Line(s) field must be set to appropriately deal with the number of lines to skip prior to reading the first line that will be transformed into objects in the repository.

5. If the user was using a **Database** as a **Datasource**, then the fields of the Database area of the dialog would be filled in, but in the case of using a csv file it is not necessary to complete any of the Database related fields.

6. Select the checkbox where it says "**For each record create an object ....**" and choose the desired QLM template from the drop-down box.

7. It is also possible to write QualiWare Command Language functions to process data and if doing so, the function name would be placed in the "**For each record call the function**" field. The details of how to write such functions will NOT be described in this user guide.

8. By selecting the **Autofill Input Choices** button, the Input Choices window will be populated with the names of the columns from the csv file.

Next step is to map the input file to QLM. To begin the mapping place your cursor in the "Fields to Attribute mapping"-window and right-click and select insert. See Figure 4
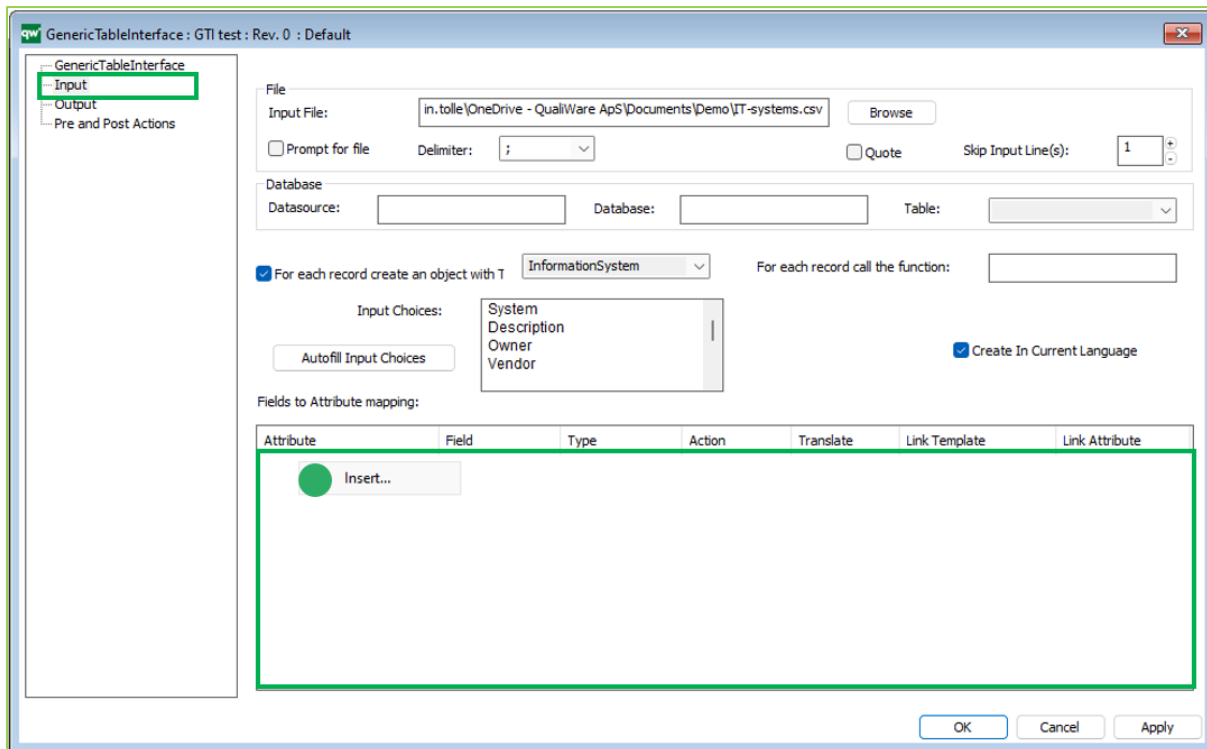
Figure 4:Add lines for mapping by right-click and select "Insert" in the "Fields to Attribute mapping"

The Attribute mapping can be set up with the following steps, cf. Figure 5:

1. Right-clicking in the Attribute Mapping field will add lines for import. When a line is added the Name is shown in the **Attribute column**. The Attribute name can be renamed by pressing F2, and a drop-down will appear with a list of Attribute associated to the Template-type selected in step 6 above, cf. Figure 3

2. By double-clicking in the **Field column** of the window (or pressing F2), the attribute list from the Input Choices window will appear. Choose the appropriate attribute to link or map to.

3. The user can then double-click in the **Type column** (or press F2), and select the appropriate type:

   a. **ID**. This implies that for each row of the spreadsheet where Field is populated, a new Object (of the selected Template-type in step 6 above) will be created in the repository with the name.

      i. When the type is "**ID**" the import routine will reuse objects with the same name instead of creating duplicate objects in the repository.

      ii. If ID is not set in the Type field, duplicate objects with the same name will be created during the import.

   b. It is also possible to choose Text, Link, LinkedBy, or Date formats, depending on what data is in the corresponding field in the csv file.

4. If there are multiple links or text fields in the csv file, the **Action column** of the Attribute Mapping window can also be used to define what sort of action it to be taken such as Overwrite, Append to List in Object, Prepend to List in Object, etc.

5. The **translate column** permits the user to specify how to format text, for instance in all lower case, upper case, or mixed case. If nothing is selected in the column, the text will be loaded just as it appeared in the cell in the csv file.

6. When the Type is set to Link, the Field can be link to another template, specified in the **Link Template column**. The Link Template field specifies what template the objects should Link to, and the type of relation between the objects is selected in the Attribute column, e.g.:

    a. In the example below the "Owner"-column of the csv file contains the Position names of the owner. The GTI would create a link (SystemOwner) from the InformationSystem-object to a Position-object with the name in the "Owner" field.

        i. If the repository and/or CSV-file already contains an instance of the selected Link Template (Position) with an identical name, the GTI would link to the existing object (Position), cf. the Action "Overwrite".

        ii. If the repository does not contain a Position-object with the identical name, the GTI will create a new Position-object, and link to that.

    b. In the example below the "Vendor"-column of the csv file contains the Vendors name. The GTI would create a link (SystemVendor) from the InformationSystem-object to a BusinessConnection-object with the name in the "Vendor" field.

        i. If the repository and/or the CVS-file contain an instance of the selected Link Template (BusinessConnection) with identical name, the GTI would reuse the existing object (BusinessConnection).

        ii. If the repository does not contain a BusinessConnection-object with the identical name, the GTI will create a new BusinessConnection-object, and link to that.

7. Similarly, when the Type is set to LinkedBy, the Field can be link to another template, specified in the Link Template column. The Link Template field will represent what template the imported objects should be LinkedBy, and the linkedby-relationship is specified in the **LinkAttribute column** (not shown in the example).
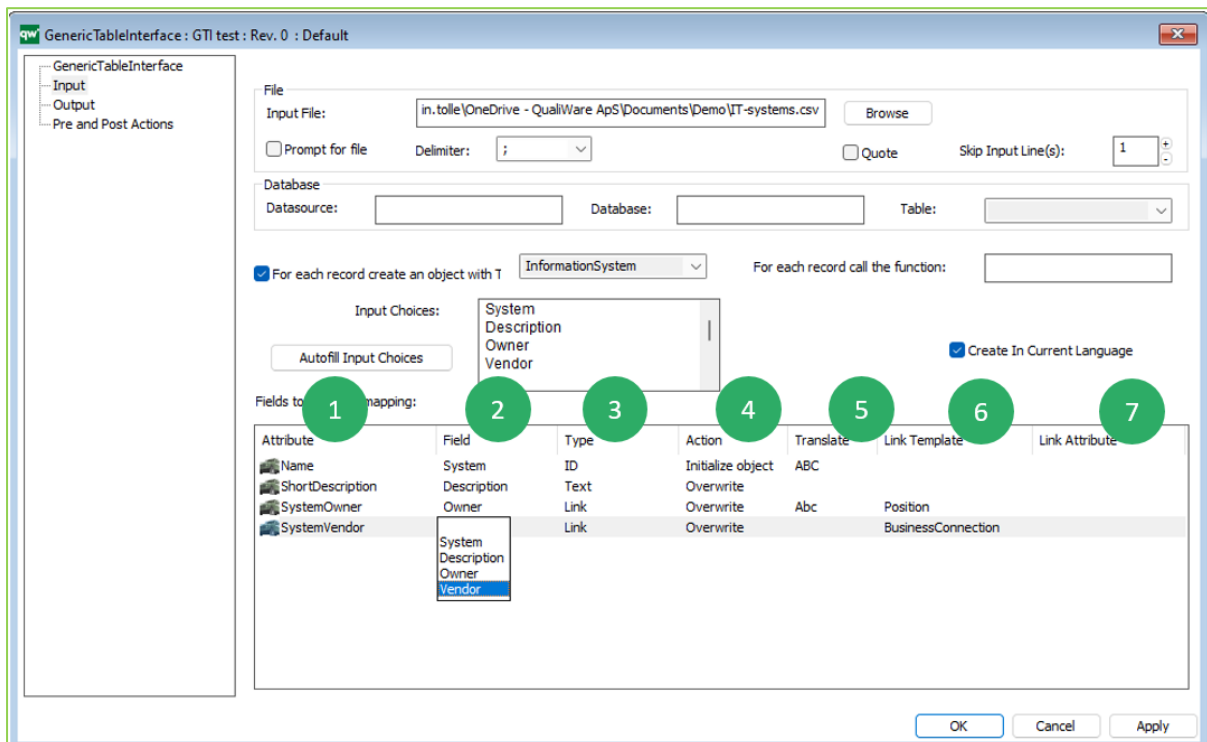

Figure 5: Setting up the Attribute mapping between import file and QLM

4.2 Export data from QLM

The GTI can also be used to export QLM repository content to a new csv file. The dialog is similar with respect to the fields of the Object Editor, and to accomplish an export the user would simply use the Output tab of the Generic Table Interface, cf. Figure 6.
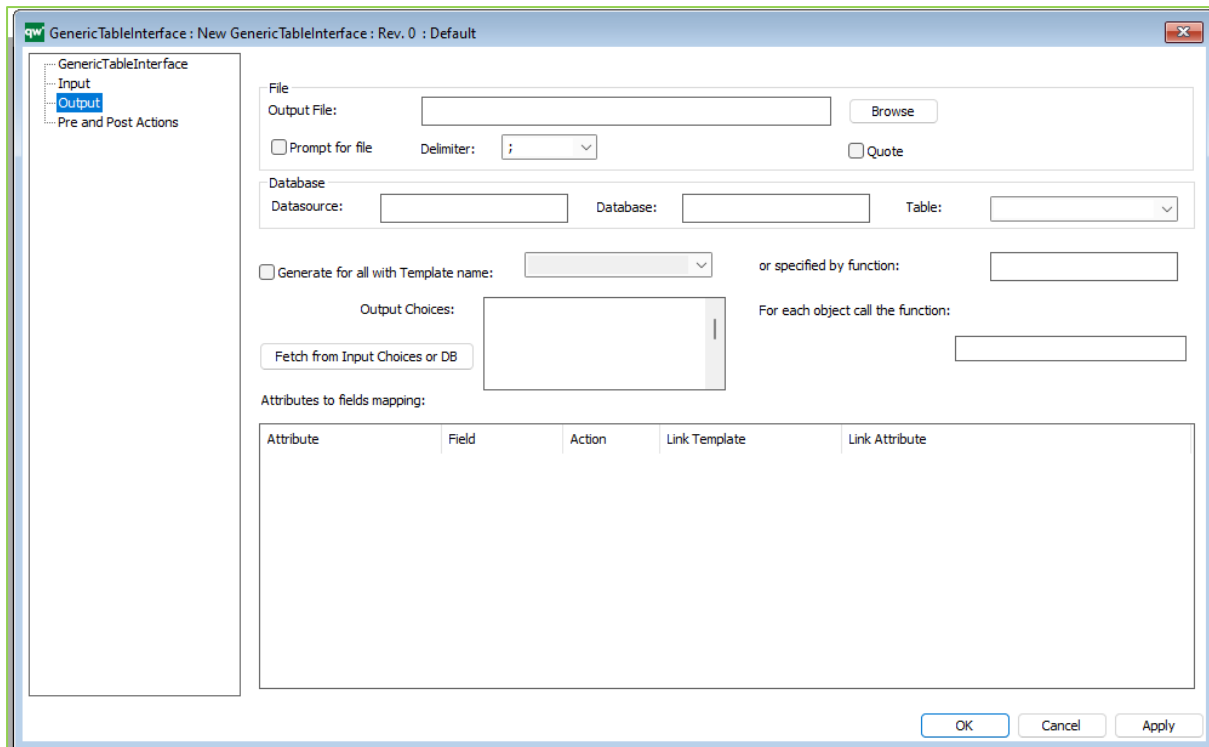

Figure 6: Output-tab of the GTI

4.3 Generate functions and execute import/export

Once the mapping is completed on either the Input or Output tab (depending on whether an Import or an Export is desired) the GTI can be executed from the GenericTableInterface tab, cf. Figure 7.
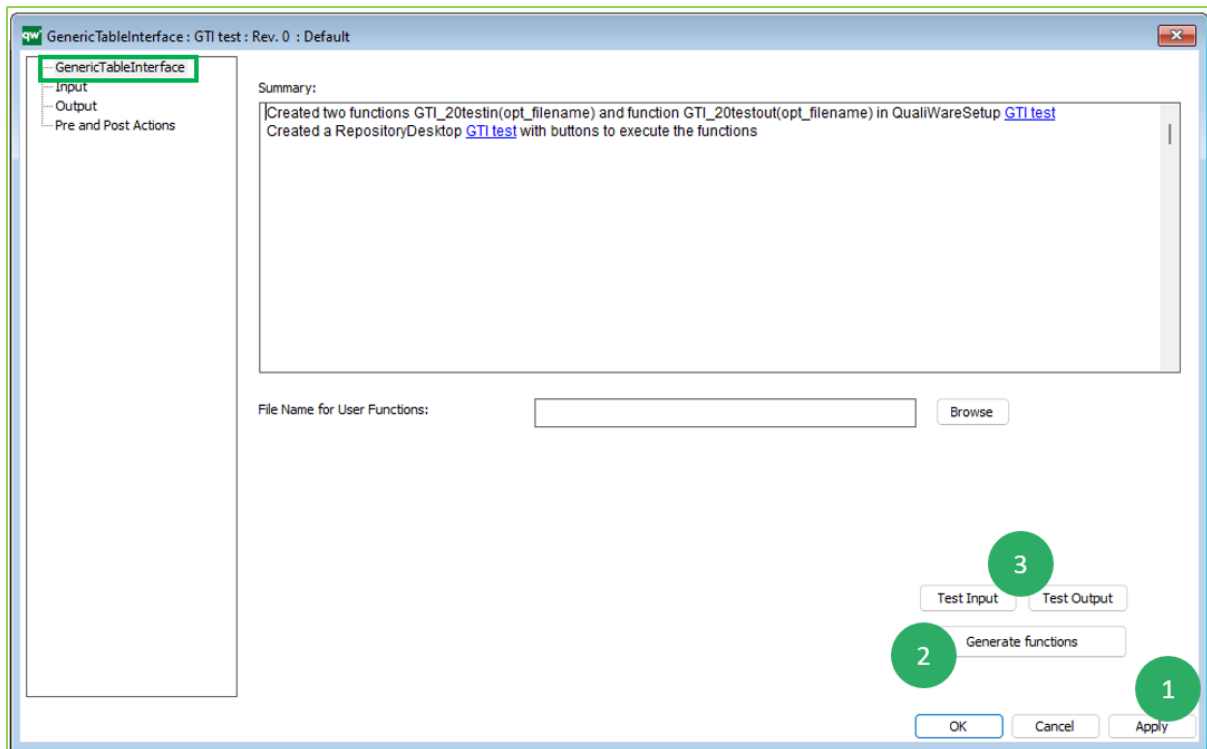
Figure 7: Generate Functions and Test Input/Output

The import/export is executed using the following steps:

1. Save the GenericTableInterface by selecting **Apply** on the lower right corner of the Object Editor dialog.
2. On the main tab (GenericTableInterface), select the Generate Functions button. This action will generate a set of QCL (QualiWare Command Language) that can be executed, and a description will be shown in the **Summary** window.
3. To execute the Import, select the **Test Input** button and the data in the csv file will be imported. To execute an Export, the user would select the **Test Output** button.

Following these actions, the user can and should verify that the data either imported or exported is as expected.

Contact your local QualiWare Support organization if there are any questions regarding the use of the Generic Table Interface.