



Altair

---

**HyperWorks**

Altair HyperMesh 2019 Tutorials

HM-8060: Calculate the Resultant Sum of Forces

## HM-8060: Calculate the Resultant Sum of Forces

In this tutorial you will create a Tcl script that:

- Determines the components of a given set of force vectors
- Calculates a vector resultant sum of the forces
- Reports the resultant sum

### Tools

---

The Tcl commands `if`, `foreach` and `expr` will be used to add logic and mathematical functions to the script. The command `hm_getentityvalue` is used to extract information from HyperMesh entities, based on data names.

Data names are generic references to the information that physically define an entity in the HyperMesh environment. An example of this is the x, y, and z coordinates that define a node location in three-dimensional space. The available data names for each entity can be found in the HyperMesh Reference Guide [Data Names](#) topic.

Data names are accessed using the `hm_getentityvalue` command. This command uses the data names available for an entity to return the particular value of interest. The command will return a value that is either a string or a numeric value, depending on the command syntax and the value stored in that particular data name field. The basic syntax of the command is:

```
hm_getentityvalue entity_type id data_name flag
```

where `entity_type` is the requested entity type (elements, loads, nodes, etc.), `id` is the entity ID, the `data_name` is the data field name of interest, and `flag` is either 0 or 1 depending on whether the command should return a numeric value (0) or a string (1).

To retrieve the x-component of a force with ID 12, the following command can be used:

```
set force_x [hm_getentityvalue loads 12 "comp1" 0]
```

**Note:** To assign the value from the command to a variable, the command is placed within square brackets.

### Exercise

---

Create a Tcl script to compute the resultant sum of a given selection of forces. This requires that the script read data from the force entities and manipulate the data to calculate the resultant. To calculate the resultant of the forces, retrieve the x, y, and z components of the forces and compute a vector sum.

1. Define the process.
2. Determine the data names to use to extract the force components.
3. Create the Tcl script and add logic as necessary.
4. Test the script.

**Step 1: Define the process.**

The script should automate the following process:

- Prompt the user to select a number of forces to calculate the resultant.
- Make sure the user has selected one or more loads.
- Extract the components of each force.
- Sum the components.
- Report the result to the user.

**Step 2: Determine the data names to use to extract the force components.**

The following table lists several relevant data names for force loads:

|                            |  |
|----------------------------|--|
| <code>comp1</code>         | x component of the vector  |
| <code>comp2</code>         | y component of the vector  |
| <code>comp3</code>         | z component of the vector  |
| <code>config</code>        | The number "1" for forces<br>The number "2" for moments<br>The number "8" for velocities<br>The number "9" for accelerations |
| <code>entitytype</code>    | the type of entity to which this load is applied (1=node, 3=component, 10=set)   |
| <code>node</code>          | when a load is applied to a node, this serves as a pointer to the node   |
| <code>inputsystemid</code> | reference system ID  |

**Steps 3-9: Create the Tcl script and add logic as necessary.**

A Tcl script to perform this function might be similar to the following:

**Step 3: Open a text file and save the file as HM8060.tcl.**

#### Step 4: Select the desired loads and then add those loads to a list.

The `*createmarkpanel` command is used to allow the user to graphically select the loads from the HyperMesh interface and add them to the mark. The command below adds the loads to mark 1. Once the loads have been added to mark 1, the load ids are assigned to a list called `loads_list`, using the TCL command `set`. Add the following 2 lines to the file `HM8060.tcl`:

```
*createmarkpanel loads 1 "Select forces to compute resultant";  
set loads_list [hm_getmark loads 1];
```

#### Step 5: Initialize the variables for the x, y, and z components.

Use the TCL command `set` to initialize the variables for the x, y, and z components of force to 0. These variables will be used later in the script to compute the resultant force for each component. Add the following 3 lines to the TCL file `HM8060.tcl`:

```
set x_comp_sum "0";  
set y_comp_sum "0";  
set z_comp_sum "0";
```

#### Step 6: Begin an `if` loop which checks to see if the variable `loads_list` has values. If it does, proceed with the macro.

Before calculating the resultant of the forces selected, we should check to make sure that the variable `loads_list` has values in it. This is done by using an `if` loop. In the `if` loop below, we are checking that the variable `loads_list` is not empty. Add the following line to the TCL file to initialize the `if` loop:

```
if {$loads_list != ""} {
```

#### Step 7: Use a `foreach` loop to iterate through each load in the list `loads_list` and extract the x, y, and z components using the `hm_getentityvalue` command and the appropriate data name.

Using a `foreach` loop, each load in the list `loads_list` will be iterated through. Within the `foreach` loop, each load is referenced by `load_id` and then the component value is added to the previous loads component's value. For example, let's look at the x component. Using the `set` command, the variable `x_comp_sum` is defined as the previous value of `x_comp_sum`, plus the x component of the current load. The x component of the current load is retrieved by using the `hm_getentityvalue` command and the data name `comp1` (all the available data names for loads are shown in the table above). This process is done for the y and z components as well. Add the following 4 lines to the TCL file:

```
foreach load_id $loads_list {  
    set x_comp_sum [expr $x_comp_sum + [hm_getentityvalue
```

```
loads $load_id "comp1" 0]];
    set y_comp_sum [expr $y_comp_sum + [hm_getentityvalue
loads $load_id "comp2" 0]];
    set z_comp_sum [expr $z_comp_sum + [hm_getentityvalue
loads $load_id "comp3" 0]];
}
```

### Step 8: Report the Resultant Force to the user.

Use the command `hm_usermessage` to report each of the components of the resultant force. Add the following line to the TCL file:

```
hm_usermessage "Resultant force: $x_comp_sum, $y_comp_sum
$z_comp_sum ";
```

### Step 9: Complete the if loop and report an error message if no loads are found.

To complete the `if` loop, add an `else` statement. Remember the `if` statement checked to see if the variable `loads_list` was not empty. This `else` statement returns an error message to the user to let them know that no loads were selected.

```
} else {
    hm_errormessage "No loads selected";
}
```

### Step 10: Test the script.

1. From the menu bar, select **View > Command Window** to display the **Command Window** at the bottom of the screen.
2. Click and drag to open the **Command Window** from the bottom edge of the screen.
3. Open the file `c_channel-tcl.hm`.
4. Use the source command to execute the script. For example:

```
source HM8060.tcl
```

It is often necessary to debug Tcl scripts using the **Command Window**. This allows you to run the Tcl script and easily review error messages, as well as print out debug information. Additional details can be found in the *Creating Tcl Scripts and Running Tcl Scripts* sections.

There are several important assumptions used when creating this script.

- The user will always select force loads, as opposed to moments, pressures, etc.
- The forces are applied to nodes, as opposed to comps or sets, and are valid to sum.
- All of the forces are applied in the same coordinate system so that it is valid to sum the component values directly.

If any of these assumptions are not true, the values returned by the script may be invalid. Additional conditional logic can be programmed to check for each of these situations and an error message can be returned or they can be handled appropriately.

5. The result of the macro is shown in the status bar. Either a message with the resultant force is shown or else there is a note saying that no loads were selected.