

Multilevel Rowsets

A multi-level rowset is one that has a parent-child-grandchild relationship structure. Such rowsets can be really handy when you need to emulate the structure of the component buffer (scroll levels).

For this example I'm going to use a structure based on PeopleTools security - the `USER_ROLES` page in the `USERMAINT` component which is located at:

PeopleTools > Security > User Profiles > User Profiles > Roles (tab)

The reason I'm using PeopleTools security is that it is available for all PeopleSoft applications (HRMS, Campus Solutions, Finance etc).

A simplified version of the scroll structure in the `USER_ROLES` page is:

```
+ Level 0: Operator ID (PSOPRDEFN)
  - Level 1: Role (PSROLEUSER_VW)
```

What I want is to create a rowset that emulates this scroll structure so that I can store the roles a particular user may have.

The first step is to declare the rowsets and use an appropriate naming convention and suffix with the scroll level which I makes sense:

```
Local Rowset &rsUser0;
Local Rowset &rsRoles1;
Local Rowset &rsUserRoles;
```

Rowset `&rsUser0` is for `PSOPRDEFN`, `&rsRoles1` is for `PSROLEUSER_VW` and `&rsUserRoles` is the multi-level rowset with the structure we want.

Now we need to create the standalone rowsets for level 0 and level 1 in our scroll structure:

```
&rsUser0 = CreateRowset(Record.PSOPRDEFN);
&rsRoles1 = CreateRowset(Record.PSROLEUSER_VW);
```

Finally we can create the multi-level rowset:

```
&rsUserRoles = CreateRowset(&rsUser0, &rsRoles1);
```

This gives us a rowset where each user (operator ID) can have one or more roles. In our example, we could then populate the standalone rowset from the component buffer like so:

```
Local Rowset &rs0;  
&rs0 = GetLevel0();  
&rs0.CopyTo(&rsUserRoles);
```

You might have a case where you want multiple children at a scroll level. Here's another example using the `PSOPRALIAS` page on the `USERMAINT` component.

The structure this time is:

```
+ Level 0: PSOPRDEFN  
  - Level 1: PSOPRALIAS  
  + Level 1: PSOPRALIASTYPE  
    - Level 2: PSORPALIASFIELD
```

Now we have three scroll levels and two children at the same scroll level (scroll 1 has `PSOPRALIAS` and `PSOPRALIASTYPE`).

Once again we start with the standalone rowsets for each record in the scroll structure. However this time, using a bottom-up approach we link the lower scroll levels to the higher scroll levels as follows:

```
Local Rowset &rsUser0, &rsAlias1, &rsAliasType1, &rsAliasField2;  
  
&rsAliasField2 = CreateRowset(Record.PSORPALIASFIELD);&rsAliasType1 =  
CreateRowset(Record.PSORPALIASTYPE, &rsAliasField2);&rsAlias1 = CreateRowset(Record.PSORPALIAS,  
&rsAliasType1);  
&rsUser0 = CreateRowset(Record.PSORPRDEFN);
```

Now to create the multilevel rowset.

```
Local Rowset &rsUserAlias;  
&rsUserAlias = CreateRowset(&rsUser0, &rsAlias1);
```

When you are referencing anything below scroll 0, you need to use the same syntax as you would when populating a rowset from the component buffer.

For example, if I wanted to populate the `PSOPRALIAS` (scroll 1) of the `&rsUserAlias` multilevel rowset with the Rowset fill method (for standalone rowsets) for just the `PS` user, the code would look like this:

```
&rsUserAlias(1).GetRowset(Scroll.PSOPRALIAS).Fill("where OPRID = 'PS'");
```

The same applies for getting to a particular field. For example, say I wanted to message out the field `PSOPRALIAS.OPRALIASTYPE`. This would be the code to use with the `&rsUserAlias` multilevel rowset:

```
MessageBox(0, "", 0, 0, "Operator Alias Type = " |  
&rsUserAlias(1).GetRowset(Scroll.PSOPRALIAS)(1).PSOPRALIAS.OPRALIASTYPE.Value);
```

Going down to level 2 or 3 can get quite confusing! That's why it is easier to do the manipulation to the underlying rowsets then combine them together at the end into a multilevel rowset.

Getting information out of the multilevel rowset still provide the same challenges though. Which is why you might want to create scroll level rowsets that reference scroll levels on your multilevel rowset. The previous examples can be simplified this way by creating `&rsUserAlias1` which references the `PSOPRALIAS` scroll at level 1 on the `&rsUserAlias` multilevel rowset.

```
Local Rowset &rsUserAlias1;  
  
&rsUserAlias1 = &rsUserAlias(1).GetRowset(Scroll.PSOPRALIAS);  
  
MessageBox(0, "", 0, 0, "Operator Alias Type = " &rsUserAlias1.PSOPRALIAS.OPRALIASTYPE.Value);
```

Remember, when it comes to rowsets, its all about context!

Revision #3

Created Mon, Sep 30, 2019 12:05 AM by [PeopleSoft Wiki](#)

Updated Mon, Sep 30, 2019 12:15 AM by [PeopleSoft Wiki](#)