Sometimes we may have requirement to concatenate values of two fields of different datatype (i.e. string type + Real type) in a single column.

To fulfil the requirement, we can create new view and can add computed column in it.

Below are the steps for reference to concatenate “Service Order ID” and “Service Order Line Number” of Service management module in a single column.

1. Create new View “GetWorkOrder”
2. Add Data source in the view. [DataSource can be Table or Query]. Name the datasource “WorkOrder”
3. Add fields “ServiceOrderId” and “ServiceOrderLineNum” in the view
4. Add computed Method for WorkOrderNumber: ServiceOrderId (8 digit)+Serviceorderlinenum (2 digit)

**private** **static** **server** **str** compWONumberMethod()

{

#define.ViewName(GetWorkOrder)

#define.DataSourceName("WorkOrder")

#define.FieldsServiceOrderId("ServiceOrderId")

#define.FieldsServiceOrderLineNum("ServiceOrderLineNum")

*//#define.FieldsServiceOrderLineNum("TELServiceObjectDescription")*

**real** rLineNum;

**str** sReturn,

sServiceOrderId,

sServiceOrderLineNum;

DictView dictView2;

*// Construct a DictView object for the present view.*

dictView2 = **new** DictView(**tableNum**(#ViewName));

*// Get a string that has the target field name*

*// propertly qualified with an alias (such*

*// as "A." or "B.").*

sServiceOrderId = dictView2.computedColumnString

(#DataSourceName,

#FieldsServiceOrderId,

FieldNameGenerationMode::FieldList,

**true**);

sServiceOrderLineNum = dictView2.computedColumnString

(#DataSourceName,

#FieldsServiceOrderLineNum,

FieldNameGenerationMode::FieldList,

**true**);

*//sServiceOrderLineNum =*

sReturn = sServiceOrderId;*//+"-"+sServiceOrderLineNum;*

sReturn = sServiceOrderId + " + + " + "REPLACE(STR(convert(varchar(10), convert(int," + sServiceOrderLineNum + ")) , 2), SPACE(1), '0')" ;

**return** sReturn;

}