

Groovy Script Type: *Action Handler*

ID	Groovy Script Name	Groovy Script Text
322	322	return "alert('Test!')"
39	Activity Script	import com.agileassetsinc.core.*; throw new TRDIEException("activity groovy script not made yet");
68	Add 0.5 to current value	Double ret; ret=val_in<4.5?5:val_in+0.5; return ret;
350	No Change	return val_in;
323	Improve Rutting by 0.13	Double ret; ret=val_in<0.13?0:val_in-0.13; return ret;
301	Grind with Slab Replacement	Double ret=342; return ret;
300	Slab replacement	Double ret=338; return ret;
299	Crack, Seat and Overlay (CSOL)	Double ret=339; return ret;
298	Grind	Double ret=343; return ret;
297	PCC lane replacement	Double ret=344; return ret;
296	Thin Overlay	Double ret=336; return ret;
295	Very Thin Overlay	Double ret=349; return ret;
294	Seal Cracks	Double ret=341; return ret;
293	Full Depth Reclamation	Double ret=340; return ret;
292	Seal Coat	Double ret=337; return ret;

Groovy Script Type: *After Treatment*

ID	Groovy Script Name	Groovy Script Text
291	Cold In-Place	Double ret=345; return ret;
290	Thick Overlay or Reconstruct	Double ret=348; return ret;
289	Medium Overlay	Double ret=347; return ret;
288	Set to Rigid	Double ret=2; return ret;
281	Set to Flexible	Double ret=1; return ret;
-42	General: No Improvement	Double ret; ret=val_in; return ret;
-18	General: Add 1.0 to current value	Double ret; ret=val_in>4?5:val_in+1; return ret;
151	PerfModel= ITD - Rigid - Surface	Double ret = 13; return ret;
129	Set to Value 3	Double ret = 3; return ret;
128	PerfModel= ITD - Rigid - 4R	Double ret = 12; return ret;
127	PerfModel= ITD - Rigid - 3R	Double ret = 11; return ret;
126	PerfModel= ITD - Rigid - 3R	Double ret = 10; return ret;
125	PerfModel= ITD - Rigid - 1R	Double ret = 9; return ret;
124	PerfModel= ITD -	Double ret = 8; return ret;
123	PerfModel= ITD -	Double ret = 7; return ret;
122	PerfModel= ITD -	Double ret = 6; return ret;

Groovy Script Type: *After Treatment*

ID	Groovy Script Name	Groovy Script Text
121	PerfModel= ITD -	Double ret = 5; return ret;
120	PerfModel= ITD -	Double ret = 4; return ret;
119	PerfModel= ITD -	Double ret = 3; return ret;
118	PerfModel= ITD -	Double ret = 2; return ret;
117	PerfModel= ITD - Flexible - 3R	Double ret = 1; return ret;
115	Reset to 40	Double ret = 40.0; return ret;
114	Reset to 0	Double ret = 0.0; return ret;
134	Increment Counter	Double ret=0; if (val_in !=null){ ret=val_in+1; } return ret;
132	Set PerfModelType for Surface Coats	import com.agileassetsinc.core.*; /* sets the performance model for surface coats based on the last known treatment performance model as defined below Current Treatment Model ID Surface Coat Model ID 2 5 1 6 3 7 4 8 5 5 6 6 7 7 8 8 9 val_in 10 13 11 val_in 12 val_in 13 13*/ double ret; if (val_in == 2){ ret = 5; } else if (val_in == 1) { ret = 6; } else if (val_in == 3) { ret = 7;

ID	Groovy Script Name	Groovy Script Text
		<pre> } else if (val_in == 4) { ret = 8; } else if (val_in == 5) { ret = 5; } else if (val_in == 6) { ret = 6; } else if (val_in == 7) { ret = 7; } else if (val_in == 8) { ret = 8; } else if (val_in == 9) { ret = val_in; } else if (val_in == 10) { ret = 13; } else if (val_in == 11) { ret = val_in; } else if (val_in == 12){ ret = val_in; } else if (val_in == 13){ ret = 13; } return ret; </pre>
131	Set to Value 2	<pre> Double ret=2; return ret; </pre>
130	Set to Value 1	<pre> Double ret=1; return ret; </pre>
113	Improve to 4.00	<pre> Double ret = 4.00; return ret; </pre>
37	Improve to 100	<pre> Double ret = 100.0; return ret; </pre>
69	Add 1.5 to current value	<pre> Double ret; ret=val_in>3.5?5:val_in+1.5; return ret; </pre>
70	Add 1.0 to current value	<pre> Double ret; ret=val_in>4?5:val_in+1; return ret; </pre>
71	Add 0.25 to current value	<pre> Double ret; ret=val_in>4.75?5:val_in+0.25; return ret; </pre>
73	Improve to 5	<pre> Double ret = 5.0; return ret; </pre>
185	Material Transfer	<pre> import com.agileassetsinc.core.*; import com.agileassetsinc.main.MaterialFunctio ns; import java.util.Set; int j = dsIn.rowCount(); for(int i=1; i<=j; i++) { if(dsIn.isRowNew(i)) { // insert </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dsIn.getItemString(i, "REQUEST_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 1, dsIn.getItemInteger(i, "MASTER_CODE_ID"), dsIn.getItemInteger(i, "OWNER_ID_FROM"), dsIn.getItemInteger(i, "OWNER_ID_TO")); } if(dsIn.getItemString(i, "REPLY_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 2, dsIn.getItemInteger(i, "MASTER_CODE_ID"), dsIn.getItemInteger(i, "OWNER_ID_FROM"), dsIn.getItemInteger(i, "OWNER_ID_TO")); } } else { // update Integer masterCodeId = dsIn.getItemInteger(i, "MASTER_CODE_ID"); Integer ownerIdFrom = dsIn.getItemInteger(i, "OWNER_ID_FROM"); Integer ownerIdTo = dsIn.getItemInteger(i, "OWNER_ID_TO"); if(dsIn.getItemStringOriginal(i,"REQUEST_USER_UPDATE")==null && dsIn.getItemString(i, "REQUEST_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 1, masterCodeId, ownerIdFrom, ownerIdTo); } if(dsIn.getItemStringOriginal(i,"REPLY_USER_UPDATE")==null && dsIn.getItemString(i, "REPLY_USER_UPDATE")!=null && ! </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO")) { handleMessages(dl, 2, masterCodeId, ownerIdFrom, ownerIdTo) ; } } def handleMessages(DataLayer dl, int messageType, Integer masterCodeId, Integer ownerIdFrom, Integer ownerIdTo) throws TRDException { MaterialFunctions mf = new MaterialFunctions(); String masterCodeName = mf.getTypeColumnNameById(dl, "MASTER_CODE_ID", masterCodeId); String title = "Request for transfer"; if(messageType == 1) { String text = "Request for transfer for material "+masterCodeName+" has been made by "+mf.getTypeColumnNameById(dl,"OWNER_ID",ownerIdTo); Set<String> userIds = SystemNotification.groupToUsers(dl, notificationTypeId, ownerIdTo); SystemNotification.send(dl, notificationTypeId, userIds, title, text); } else if(messageType == 2){ String text = "Request for transfer for material "+masterCodeName+" has been fulfilled by "+mf.getTypeColumnNameById(dl,"OWNER_ID",ownerIdFrom); Set<String> userIds = SystemNotification.groupToUsers(dl, notificationTypeId, ownerIdFrom); SystemNotification.send(dl, notificationTypeId, userIds, title, text); } } </pre>
-39	General: Material Transfer	<pre> import com.agileassetsinc.core.*; import com.agileassetsinc.main.MaterialFunctions; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> int j = dsIn.rowCount(); for(int i=1; i<=j; i++) { if(dsIn.isRowNew(i)) { // insert if(dsIn.getItemString(i, "REQUEST_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 1, dsIn.getItemInteger(i, "MASTER_CODE_ID"), dsIn.getItemInteger(i, "OWNER_ID_FROM"), dsIn.getItemInteger(i, "OWNER_ID_TO")) ; } if(dsIn.getItemString(i, "REPLY_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 2, dsIn.getItemInteger(i, "MASTER_CODE_ID"), dsIn.getItemInteger(i, "OWNER_ID_FROM"), dsIn.getItemInteger(i, "OWNER_ID_TO")) ; } } else { // update Integer masterCodeId = dsIn.getItemInteger(i, "MASTER_CODE_ID"); Integer ownerIdFrom = dsIn.getItemInteger(i, "OWNER_ID_FROM"); Integer ownerIdTo = dsIn.getItemInteger(i, "OWNER_ID_TO"); if(dsIn.getItemStringOriginal(i,"REQUEST_USER_UPDATE")!=null && dsIn.getItemString(i, "REQUEST_USER_UPDATE")!=null && !dsIn.getItemInteger(i,"OWNER_ID_FROM").equals(dsIn.getItemInteger(i,"OWNER_ID_TO"))) { handleMessages(dl, 1, masterCodeId, ownerIdFrom, ownerIdTo) ; } } if(dsIn. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getItemStringOriginal(i,"REPLY_USER_ UPDATE")==null && dsln.getItemString(i, "REPLY_USER_UPDATE")!=null && !dsln.getItemInteger(i,"OWNER_ID_FR OM").equals(dsln.getItemInteger(i,"OW NER_ID_TO")))) { handleMessages(dl, 2, masterCodeId, ownerIdFrom, ownerIdTo) ; } } } def handleMessages(DataLayer dl, int messageType, Integer masterCodeId, Integer ownerIdFrom, Integer ownerIdTo) throws TRDException { MaterialFunctions mf = new MaterialFunctions(); String masterCodeName = mf.getTtypeColumnNameById(dl, "MASTER_CODE_ID", masterCodeId); String title = "Request for transfer"; if(messageType == 1) { String text = "Request for transfer for material "+masterCodeName+" has been made by "+mf.getTtypeColumnNameById(dl,"OW NER_ID",ownerIdTo); Set<String> userIds = SystemNotification.groupToUsers(dl, notificationTypeId, ownerIdTo); SystemNotification.send(dl, notificationTypeId, userIds, title, text); } else if(messageType == 2){ String text = "Request for transfer for material "+masterCodeName+" has been fulfilled by "+mf.getTtypeColumnNameById(dl,"OW NER_ID",ownerIdFrom); Set<String> userIds = SystemNotification.groupToUsers(dl, notificationTypeId, ownerIdFrom); SystemNotification.send(dl, notificationTypeId, userIds, title, text); } } } </pre>

ID	Groovy Script Name	Groovy Script Text
149	set_labor_id	<pre> import com.agileassetsinc.core.*; String sql; sql = "select get_id_emp_id() as LABOR_ID from dual"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); if(ds.getItemString(1,"LABOR_ID")==null) { throw new TRDException("Your User ID is not associated with your Employee ID. Please contact your supervisor."); } dsIn.setItemString(1,"LABOR_ID",ds.get ItemString(1,"LABOR_ID")); </pre>
195	FLEET_Set_EQ_Attributes	<pre> import com.agileassetsinc.core.*; //return "dw_1.SetItem(1, 'MAKE', ds_equipment_details.GetItem(1,'MAKE'));"; String eqname = "ds_equipment_details.GetItem(1,'EQUI PMENT_NAME')"; String sql; sql = "select VIN, MAKE, MODEL, PROD_YEAR from EQUIPMENT_INVENTORY where EQUIPMENT_NAME = '" + eq_name + "'"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); dsIn.setItemString(1,"COMMENT_STR", "Test"); dsIn.setItemString(1,"VIN",ds.getItemStri ng(1,"VIN")); dsIn.setItemString(1,"MAKE",ds.getItem String(1,"MAKE")); dsIn.setItemString(1,"MODEL",ds.getltem String(1,"MODEL")); dsIn.setItemString(1,"PROD_YEAR",ds. getItemString(1,"PROD_YEAR")); </pre>
168	ITD_Request_User	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for ITD Requests */ String QA_CHANGE_STR; dsIn.setItemDate(1,"REQ_DATE",new DateTime()); //Set to today dsIn.setItemString(1,"LOGGED_BY", dl.getSession().getUserId()); </pre>

ID	Groovy Script Name	Groovy Script Text
166	set contract name to id	<pre>//Current user dsIn.setItemString(1,"Comments", "Notes"); import com.agileassetsinc.core.*; dsIn.setItemInteger(1,"CNTRCT_NAME", dsIn.getItemInteger(1,"CNTRCT_ID"));</pre>
161	SET_INV_ITEM_DATE	<pre>import com.agileassetsinc.core.*; String sql; sql = "select getdate() as WORK_DATE from dual"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); dsIn.setItemString(1,"DATE_WORK",ds. getItemString(1,"WORK_DATE"));</pre>
234	Negative Index	<pre>for (int i=1;i<=ds_in.rowCount();i++) { ds_in.setItemNumber(i,"PRIORITY", - ds_in.getItemNumber(i,"CA_IRI_AVG")); }</pre>
230	Weight 1 for each section	<pre>for (int i=1;i<=ds_in.rowCount();i++) ds_in.setItemNumber(i,"INDEX_WEIGHT",1.0);</pre>
233	Lane Miles	<pre>for (int i=1;i<=ds_in.rowCount();i++) { Integer id_greek_struc_id=ds_in.getItemInteger(i,"ID_GREEK_STRUC_ID"); Double coef=1.0; if(id_greek_struc_id.equals(new Integer(2))) coef=1.2; if(id_greek_struc_id.equals(new Integer(3))) coef=1.1; ds_in.setItemNumber(i,"INDEX_WEIGHT",coef*ds_in.getItemNumber(i,"LANE_MILES")); }</pre>
339	Update Cost to Scenario Results	<pre>return dsIn.rowCount() > 0</pre>
100	BeforeSaveProfiler	<pre>import com.agileassetsinc.core.* /* Script to check to see if IRI Value Has been Adjusted then Comments and Reason Code cannot be null */ for(int i=1; i<=dsIn.rowCount());</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> i++) { if (!dsIn.isItemNull(i,"ID_IRI_A_E_ADJ") && (dsIn.isItemNull(i,"ID_QA_PROF_CODE_ID") dsIn.isItemNull(i,"ID_QA_REVW_COMMENTS")))) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException(1254); //Please enter a adjustment reason code and comment for each record where the IRI value has been adjusted. } /* Script to check to see if QA review is requested then the QA Requested Comments must not be null and if QA Review is "reviewed" then QA Review comment cannot be null. */ if (dsIn.getItemNumber(i,"ID_QA_REVIEW") ==1 && dsIn.getItemString(i,"ID_QA_REQUEST_COMMENTS") ==null) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException(1245); //Please enter requestor comments for each record where QA review is requested } if (dsIn.getItemNumber(i,"ID_QA_REVW_STATUS_ID") ==3 && dsIn.getItemString(i,"ID_QA_REVW_COMMENTS") ==null) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException(1246); //Please enter QA review comments for each record that has been QA reviewed } } import com.agileassetsinc.core.*; DataStore ds = dl.createRetrieveDataStore("select * from equipment_inventory where equipment_status = 893 and plate like '?'", null, dsIn.getItemString(i,"PLATE")); if (dsIn.rowCount() > 0) {throw new TRDException("Cannot save duplicate plate value.");} </pre>
-24	General: Check_For_Duplicate_Plates	

ID	Groovy Script Name	Groovy Script Text
184	e_time_validation_before_save	<pre> import com.agileassetsinc.core.*; String workDate; Double rem; String sql; String esql; String tsq; String lsq; String trc_sql; for(int i=1; i<=dsIn.rowCount(); i++) { workDate = TRDIUtils.dateToStr(dsIn.getItemDate(1, "DATE_WORK"),"[shortdate]"); rem = dsIn.getItemNumber(1,"TOTAL_HOURS ") % 0.5; sql = "select case when " + dsIn.getItemNumber(1,"WORK_ORDER _ID") + " NOT IN (select WORK_ORDER_ID from ID_ADV_WORK_ORDER_CHRTFLD) then 1 else 0 end AS WO_CHECK," + " case when " + dsIn.getItemNumber(1,"WORK_ORDER _ID") + " IN (select WORK_ORDER_ID from ID_ADV_WORK_ORDER_CHRTFLD) then " + " case when (select ADV_PROG_CD from ID_ADV_WORK_ORDER_CHRTFLD where WORK_ORDER_ID = " + dsIn.getItemNumber(1,"WORK_ORDER _ID") + ") IS NOT NULL then 1 " + " else 0 end end AS PROG_CHECK" + " from dual "; DataStore ds = dl.createRetrieveDataStore(sql); esql = "select case when " + dsIn.getItemNumber(1,"LABOR_ID") + " > 0 then SUBSTR(ADV_SUB_TITLE_CD,0,1) " + " else 'N' end AS FLSA, LAST_NAME AS NAME " + " from LABOR_INVENTORY where LABOR_ID = " + dsIn.getItemNumber(1,"LABOR_ID"); DataStore dse = dl.createRetrieveDataStore(esql); tsq = "select SUBSTR(TRC_NAME,0,INSTR(TRC_N AME,'-')-2) AS TRC_NAME from SETUP_TRC where </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> TRC_ID = " + dsln.getItemNumber(1,"TRC_ID"); DataStore dst = dl.createRetrieveDataStore(tsql); lsql = " " + " select " + " (select max(date_work) from work_orders_labor_dc where labor_id = a.labor_id and date_work = a.date_work and approved = 1) as date_work, " + " (select max(labor_id) from work_orders_labor_dc where labor_id = a.labor_id and date_work = a.date_work and approved = 1) as labor_id, " + " nvl((select sum(total_hours) from work_orders_labor_dc where labor_id = a.labor_id and date_work = a.date_work and approved = 1 and trc_id <> " + dsln.getItemNumber(1,"TRC_ID") + "), 0) as total_hours " + " from work_orders_labor_dc a " + " where a.labor_id = " + dsln.getItemNumber(1,"LABOR_ID") + " and a.date_work = to_date(" + workDate + "','MM/dd/yyyy') " + " group by a.date_work, a.labor_id"; DataStore dsl = dl.createRetrieveDataStore(lsql); trc_sql = " " + " select " + " case when exists(select 1 from work_orders_labor_dc " + " where labor_id = " + dsln.getItemNumber(1,"LABOR_ID") + " and date_work = to_date(" + workDate + "','MM/dd/yyyy') " + " and trc_id = 14) then 1 " + " else 0 " + " end as trc_check_1, " + " case when exists(select 1 from work_orders_labor_dc " + " where labor_id = " + dsln.getItemNumber(1,"LABOR_ID") + " and date_work = to_date(" + workDate + "','MM/dd/yyyy') " + " and trc_id < 11) then 1 " + " else 0 " + " end as trc_check_2 " + " from dual "; DataStore dstrc = dl.createRetrieveDataStore(trc_sql); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> // Check whether the Work Order is saved. if(ds.getItemNumber(1,"WO_CHECK") == 1) { dl.getSession().setErrorDetails(dsln.getl d()+";"+dsln.getRowId(i)); throw new TRDException("Please save the Work Order before proceeding further."); } // Check whether the employee is selected. (For Work Order/Day Card window) if(dsln.getItemNumber(1,"LABOR_ID")= =null) { dl.getSession().setErrorDetails(dsln.getl d()+";"+dsln.getRowId(i)); throw new TRDException("Please select Employee before proceeding further."); } // E-time 2nd validation: Whether a particular TRC can be used with the Activity on the Work Order. if(ds.getItemNumber(1,"WO_CHECK") == 0 && ds.getItemNumber(1,"PROG_CHECK") == 1 && (dsln.getItemNumber(1,"TRC_ID") > 13 && dsln.getItemNumber(1,"TRC_ID") < 39)) { dl.getSession().setErrorDetails(dsln.getl d()+";"+dsln.getRowId(i)); throw new TRDException("For the Work Order # "+ dsln.getItemInteger(1,"WORK_ORDER_ ID") + ", the use of TRC/Event Code '"+ dst.getItemString(1,"TRC_NAME")+ "' is prohibited. (Check the record for employee '"+ dse.getItemString(1,"NAME")+"'"); } if(ds.getItemNumber(1,"WO_CHECK") == 0 && ds.getItemNumber(1,"PROG_CHECK") == 0 && (dsln.getItemNumber(1,"TRC_ID") <= 13 dsln.getItemNumber(1,"TRC_ID") >= 39)) { dl.getSession().setErrorDetails(dsln.getl </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> dst.getItemString(1,"TRC_NAME")+ "" is prohibited. (Check the record for employee '"+ dse.getItemString(1,"NAME")+"'"); } // E-time 1st Validation: Whether the employee can select a particular TRC based on his/her FLSA Code (ADV_SUB_TITLE_CD in LABOR_INVENTORY table). // Check for FLSA code = 'C' if(dse.getItemString(1,"FLSA")== 'C' && (dsln.getItemNumber(1,"TRC_ID")==12 dsln.getItemNumber(1,"TRC_ID")==15)) { dl.getSession().setErrorDetails(dsln.getl d()+";"+dsln.getRowId(i)); throw new TRDException("The employee '"+ dse.getItemString(1,"NAME")+"' does not have privileges to use the '"+ dst.getItemString(1,"TRC_NAME")+ "' TRC. (See labor day card for Work Order # "+ dsln.getItemInteger(1,"WORK_ORDER_ ID")+"'"); } // Check for FLSA code = 'A', 'I', 'P' if((dse.getItemString(1,"FLSA")== 'A' dse.getItemString(1,"FLSA")== 'I' dse.getItemString(1,"FLSA")== 'P') && (dsln.getItemNumber(1,"TRC_ID")==11 dsln.getItemNumber(1,"TRC_ID")==13 dsln.getItemNumber(1,"TRC_ID")==15 dsln.getItemNumber(1,"TRC_ID")==32 dsln.getItemNumber(1,"TRC_ID")==2 dsln.getItemNumber(1,"TRC_ID")==3 dsln.getItemNumber(1,"TRC_ID")==5 dsln.getItemNumber(1,"TRC_ID")==6 dsln.getItemNumber(1,"TRC_ID")==8 dsln.getItemNumber(1,"TRC_ID")==9 dsln.getItemNumber(1,"TRC_ID")==10 dsln.getItemNumber(1,"TRC_ID")==39 dsln.getItemNumber(1,"TRC_ID")==40 dsln.getItemNumber(1,"TRC_ID")==41 dsln.getItemNumber(1,"TRC_ID")==42 dsln.getItemNumber(1,"TRC_ID")==43 dsln.getItemNumber(1,"TRC_ID")==44 dsln.getItemNumber(1,"TRC_ID")==45 dsln.getItemNumber(1,"TRC_ID")==46)) { dl.getSession().setErrorDetails(dsln.getl </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getRowId(i)); throw new TRDIEException("The employee '"+ dse.getItemString(1,"NAME")+"' does not have privileges to use the '"+ dst.getItemString(1,"TRC_NAME") +"' TRC. (See labor day card for Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID")+")"); } // Check for FLSA code = 'E' if(dse.getItemString(1,"FLSA")==='E' && (dsIn.getItemNumber(1,"TRC_ID")==11 dsIn.getItemNumber(1,"TRC_ID")==13 dsIn.getItemNumber(1,"TRC_ID")==17 dsIn.getItemNumber(1,"TRC_ID")==32 dsIn.getItemNumber(1,"TRC_ID")==2 dsIn.getItemNumber(1,"TRC_ID")==3 dsIn.getItemNumber(1,"TRC_ID")==5 dsIn.getItemNumber(1,"TRC_ID")==6 dsIn.getItemNumber(1,"TRC_ID")==8 dsIn.getItemNumber(1,"TRC_ID")==9 dsIn.getItemNumber(1,"TRC_ID")==10 dsIn.getItemNumber(1,"TRC_ID")==12 dsIn.getItemNumber(1,"TRC_ID")==28 dsIn.getItemNumber(1,"TRC_ID")==29 dsIn.getItemNumber(1,"TRC_ID")==30 dsIn.getItemNumber(1,"TRC_ID")==39 dsIn.getItemNumber(1,"TRC_ID")==40 dsIn.getItemNumber(1,"TRC_ID")==41 dsIn.getItemNumber(1,"TRC_ID")==42 dsIn.getItemNumber(1,"TRC_ID")==43 dsIn.getItemNumber(1,"TRC_ID")==44 dsIn.getItemNumber(1,"TRC_ID")==45 dsIn.getItemNumber(1,"TRC_ID")==46)) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDIEException("The employee '"+ dse.getItemString(1,"NAME")+"' does not have privileges to use the '"+ dst.getItemString(1,"TRC_NAME") +"' TRC. (See labor day card for Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID")+")"); } // E-time 3rd Validation: Maximum value check for hours. // Default max value = 24 </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dsIn.getItemNumber(1,"TOTAL_HOURS") > 24 && (dsIn.getItemNumber(1,"TRC_ID") != 14 dsIn.getItemNumber(1,"TRC_ID") != 33 dsIn.getItemNumber(1,"TRC_ID") != 34 dsIn.getItemNumber(1,"TRC_ID") != 35 dsIn.getItemNumber(1,"TRC_ID") != 36 dsIn.getItemNumber(1,"TRC_ID") != 37 dsIn.getItemNumber(1,"TRC_ID") != 38)) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException("For the Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID") +" and the employee "+ dse.getItemString(1,"NAME")+ ", maximum time allowed is 24 hours."); } // Holiday max value = 8 if(dsIn.getItemNumber(1,"TOTAL_HOU RS") > 8 && dsIn.getItemNumber(1,"TRC_ID") == 14) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException("For the Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID") +" and the employee "+ dse.getItemString(1,"NAME")+ ", maximum time allowed for holiday is 8 hours."); } // Other exceptions: max value = 10 if(dsIn.getItemNumber(1,"TOTAL_HOU RS") > 10 && (dsIn.getItemNumber(1,"TRC_ID") == 24 dsIn.getItemNumber(1,"TRC_ID") == 33 dsIn.getItemNumber(1,"TRC_ID") == 34 dsIn.getItemNumber(1,"TRC_ID") == 35 dsIn.getItemNumber(1,"TRC_ID") == 36 dsIn.getItemNumber(1,"TRC_ID") == 37 dsIn.getItemNumber(1,"TRC_ID") == 38)) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException("For the Work Order # "+ </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> selected TRC is 10 hours."); } // E-time 4th validation: time interval check (default is half hour increment) if(rem != 0 && dsIn.getItemNumber(1,"TRC_ID") != 32) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException("For the Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID") +" and the employee "+ dse.getItemString(1,"NAME")+ ", please enter time in half hour increment."); } // Note that the 5th validation of minimum time entered states that for CPP min time should be 0.1 hour and for rest min time should be 0.5 hour. The TOTAL_HOURS field is set, at the database level and in SETUP_COLUMN_ID, to accept values in up to one decimal point. This covers the min time entry of 0.1 for CPP. The half hour interval check covers the min time entry of 0.5 for rest of the TRCs. // E-time 6th validation: no more than 24 hours of approved time in a day. if(((dsl.getItemNumber(1,"TOTAL_HOU RS") + dsIn.getItemNumber(1,"TOTAL_HOURS ")) > 24) && dsIn.getItemNumber(1,"APPROVED") == 1) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException("Cannot continue with approval of day cards for the Work Order # "+ dsIn.getItemInteger(1,"WORK_ORDER_ ID") +" and the employee "+ dse.getItemString(1,"NAME")+ " because the total hours for the employee will exceed 24 hours for " + workDate + "."); } // E-time 7th validation: Cannot use HOL and REG/OJT/OJTL type codes in same day. if((dstc.getItemNumber(1,"TRC_CHEC K_1") == 1 && </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ==45 dsln.getItemNumber(1,"TRC_ID")==46)) (dstrc.getItemNumber(1,"TRC_CHECK_ 2") == 1 && dsln.getItemNumber(1,"TRC_ID") == 14)) { dl.getSession().setErrorDetails(dsln.getI d()+";"+dsln.getRowId(i)); throw new TRDException("An employee cannot have holiday hours and work hours for the same day. Check data entry for the Work Order # "+ dsln.getItemInteger(1,"WORK_ORDER_ ID")+" and the employee "+ dse.getItemString(1,"NAME")+ " for " + workDate + "."); } </pre>
152	check_mwp_ownership_before_save	<pre> import com.agileassetsinc.core.*; import com.agileassetsinc.main.MasterPlanFun ctions; //script used for editing rows in the master work program. String sql_str_owner; String sql_str; for(int i=1; i<=dsln.rowCount(); i++) { sql_str="select count(*) as CNT "+ " FROM setup_loc_ident t1, setup_loc_ident t2, PMS_MASTER_WP p1, PMS_MASTER_WP p2 " + " WHERE t1.sourse_table = 'PMS_MASTER_WP' and p1.PMS_MASTER_WP_TYPE_ID=p2.P MS_MASTER_WP_TYPE_ID " + " AND t2.sourse_table = 'PMS_MASTER_WP' " + " AND p1.plan_year = p2.plan_year " + " AND t1.route_id = t2.route_id " + " AND p1.loc_ident=t1.loc_ident " + " AND p2.loc_ident=t2.loc_ident " + " AND (t1.lane_dir = t2.lane_dir OR t1.lane_dir = 0 OR t2.lane_dir = 0)" + " AND (t1.lane_id = t2.lane_id OR t1.lane_id = 0 OR t2.lane_id = 0)" + " AND t2.offset_from < t1.offset_to" + " AND t2.offset_to > t1.offset_from" + " AND t1.loc_ident<>t2.loc_ident"+ " AND t2.loc_ident=" + dsln.getItemInteger(i,"LOC_IDENT") ; DataStore ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")!=0) { dl.getSession().setErrorDetails(dsln. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getld()+" "+dsln.getRowId(i)); throw new TRDException("The segment overlaps another segment. Please check the milepoints"); } } ArrayList<Location> locs = new ArrayList<Location>(); for(int i = 1; i<=dsln.rowCount(); i++) { if (dsln.isLocationModified(i) dsln.isItemModified(i, "PLAN_YEAR") dsln.isItemModified(i, "PMS_TREATMENT_ID")) { locs.add(dsln.getLocation(i)); } } if (locs.size() > 0) { MasterPlanFunctions.checkTestSections (dl, locs, "PMS_CA_LTTP_TST_SECTIONS", "CA_LTTP_TEST_SECTION_REVIEW", "CHANGE IN MWP"); } </pre>
143	Script for avoiding overlapping	<pre> import com.agileassetsinc.core.*; String sql_str; for(int i=1; i<=dsln.rowCount(); i++) { sql_str="select count(*) as CNT "+ " FROM setup_loc_ident t1, setup_loc_ident t2, PMS_ID_CLASS_DATA p1, PMS_ID_CLASS_DATA p2 " + " WHERE t1.sourse_table = 'PMS_ID_CLASS_DATA' " + " AND t2.sourse_table = 'PMS_ID_CLASS_DATA' " + " AND t1.route_id = t2.route_id " + " AND p1.loc_ident=t1.loc_ident " + " AND p2.loc_ident=t2.loc_ident " + " AND (t1.lane_dir = t2.lane_dir OR t1.lane_dir = 0 OR t2.lane_dir = 0)" + " AND (t1.lane_id = t2.lane_id OR t1.lane_id = 0 OR t2.lane_id = 0)" + " AND t2.offset_from < t1.offset_to" + " AND t2.offset_to > t1.offset_from" + " AND t1.loc_ident<>t2.loc_ident"+ " AND t2.loc_ident=" + dsln.getItemInteger(i,"LOC_IDENT") ; DataStore ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")!=0) { dl.getSession().setErrorDetails(dsln.getl d()+" "+dsln.getRowId(i)); throw new TRDException("The </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> segment overlaps another segment. Please check the milepoints"); } } </pre>
135	Duplicate Boring Record	<pre> import com.agileassetsinc.core.* String sql_str; for(int i=1; i<=dsIn.rowCount(); i++) { sql_str="select count(*) as CNT "+ " FROM setup_loc_ident t1, setup_loc_ident t2, PMS_ID_BORING_LOGS t3" + " WHERE t1.sourse_table = 'PMS_ID_BORING_LOGS'" + " AND t2.sourse_table = 'PMS_ID_BORING_LOGS'" + " AND t1.route_id = t2.route_id " + " AND (t1.lane_dir = t2.lane_dir OR t1.lane_dir = 0 OR t2.lane_dir = 0)" + " AND (t1.lane_id = t2.lane_id OR t1.lane_id = 0 OR t2.lane_id = 0)" + " AND t2.offset_from <= t1.offset_to" + " AND t2.offset_to >= t1.offset_from" + " AND t1.loc_ident<>t2.loc_ident"+ " AND t2.loc_ident=" + dsIn.getItemInteger(i,"LOC_IDENT")+ " AND t3.loc_ident = t1.loc_ident" + " AND t3.ID_KEY_NUM =" + dsIn.getItemInteger(i,"ID_KEY_NUM"); DataStore ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")>0) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException("A Record Already Exists For This Boring Log"); } } </pre>
110	checkQAColumnsNotNull	<pre> import com.agileassetsinc.core.* /* Script to check to see if Ride Index Value Has been Adjusted then USER, date, Comments and Reason Code cannot be null */ for(int i=1; i<=dsIn.rowCount(); i++) { if (!dsIn.isItemNull(i,"ID_IRI_A_E_ADJ") && </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (dsIn.isItemNull(i,"ID_PM_QA_COM") dsIn.isItemNull(i,"ID_PM_QA_DATE") dsIn.isItemNull(i,"ID_PM_QA_USER") dsIn.isItemNull(i,"ID_QA_PROF_CODE_ ID"))) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException("Please enter a adjustment reason code and comment, USER and Date for each record where the IRI value has been adjusted."); } } </pre>
90	BeforeSaveIDtraffic	<pre> import com.agileassetsinc.core.* /* Script to check to see if QA review is requested then the QA Requested Comments must not be null and if QA Review is "reviewed" then QA Review comment cannot be null. */ for(int i=1; i<=dsIn.rowCount(); i++) { if (dsIn.getItemNumber(i,"ID_QA_REVIEW ")==1 && dsIn.getItemString(i,"ID_QA_REQUEST _COMMENTS")==null) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException(1245); //Please enter requestor comments for each record where QA review is requested } if (dsIn.getItemNumber(i,"ID_QA_REVW_ STATUS_ID")==3 && dsIn.getItemString(i,"ID_QA_REVW_CO MMENTS")==null) { dl.getSession().setErrorDetails(dsIn.getI d()+";"+dsIn.getRowId(i)); throw new TRDException(1246); //Please enter QA review comments for each record that has been QA reviewed } } </pre>
77	RFP Approved By Check	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"RFP_APPRO VED")==1 && dsIn.getItemObject(1," </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> RFP_APPROVED_BY") == null) { throw new TRDException("This RFP is approved. Please enter data in the *Approved By* field."); } else return; import com.agileassetsinc.core.* /* Script to check to see if Crack Index Value Has been Adjusted then Comments and Reason Code cannot be null */ for(int i=1; i<=dsIn.rowCount(); i++) { if (!dsIn.isItemNull(i,"ID_ASC_CRACK_IN D_QA") && (dsIn.isItemNull(i,"ID_QA_FIELD_CODE _ID") dsIn.isItemNull(i,"ID_QA_REVW_COMM ENTS")))) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException(1255); //Please enter a adjustment reason code and comment for each record where the IRI value has been adjusted. } if (dsIn.getItemNumber(i,"ID_QA_REVIEW ")==1 && dsIn.getItemString(i,"ID_QA_REQUEST _COMMENTS")==null) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException(1245); //Please enter requestor comments for each record where QA review is requested } if (dsIn.getItemNumber(i,"ID_QA_REVW_ STATUS_ID")==3 && dsIn.getItemString(i,"ID_QA_REVW_CO MMENTS")==null) { dl.getSession().setErrorDetails(dsIn.getI d()+"-"+dsIn.getRowId(i)); throw new TRDException(1246); //Please enter QA review comments for each record that has been QA reviewed } } import com.agileassetsinc.core.*; DataStore dsInsert = dl.</pre>
101	BeforSaveFieldRecorder	
42	Approve/Disapprove WP	

ID	Groovy Script Name	Groovy Script Text
		<pre> createDataStore("SELECT\n" + "a.plan_year AS eff_fiscal_year,\n" + "a.activity_id,\n" + "a.comment_str,\n" + "a.total_cost,\n" + "a.inv_elem_id,\n" + "a.owner_id,\n" + "a.route_id,\n" + "a.lane_dir,\n" + "0 AS lane_id,\n" + "a.offset_from,\n" + "a.offset_to,\n" + "2 AS asset_type_id,\n" + "2 AS status,\n" + "(SELECT MIN (sp.project_id) FROM setup_project sp, project_activity_plan pac WHERE sp.module_id = 3 AND sp.owner_id = a.owner_id AND sp.status = 4 AND sp.contract_plan_type = 0 AND pac.activity_id = a.activity_id AND sp.project_id = pac.project_id) AS project_id,\n" + "b.unit_cost AS rate,\n" + "b.man_hours,\n" + "b.num_men,\n" + "ROUND (CASE WHEN b.unit_cost IS NULL OR b.unit_cost = 0 THEN NULL ELSE a.total_cost / b.unit_cost END,2) AS amount\n" + "FROM (\n" + " SELECT\n" + " a.plan_year,\n" + " a.pms_treatment_id,\n" + " c.activity_id,\n" + " b.route_id,\n" + " GREATEST (b.lane_dir, lb.lane_dir) AS lane_dir,\n" + " b.lane_id,\n" + " GREATEST (b.offset_from, lb.offset_from) AS offset_from,\n" + " 'Route: ' (SELECT sr.route_name FROM setup_route sr WHERE sr.route_id = b.route_id) ' Dir:' (SELECT sl.lane_dir_name FROM setup_lane_dir sl WHERE sl.lane_dir = GREATEST (b.lane_dir, lb.lane_dir)) ' from ' TO_CHAR (GREATEST (b.offset_from, lb.offset_from)) ' to ' TO_CHAR (LEAST (b.offset_to, lb.offset_to)) AS comment_str,\n" + " a.project_price\n" + " * CASE\n" + " WHEN b.lane_dir = 0 AND lb.lane_dir > 0\n" </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> + " THEN 0.5\n" + " ELSE 1\n" + " END\n" + " * CASE\n" + " WHEN b.offset_to > b.offset_from\n" + " THEN (LEAST (b.offset_to, lb.offset_to)\n" + " - GREATEST (b.offset_from, lb.offset_from)\n" + ")\n" + " / (b.offset_to - b.offset_from)\n" + " ELSE 1\n" + " END AS total_cost,\n" + " LEAST (b.offset_to, lb.offset_to) AS offset_to,\n" + " i.section_id AS inv_elem_id,\n" + " i.crew_id AS owner_id\n" + " FROM pms_master_wp a,\n" + " setup_loc_ident b,\n" + " setup_activity_tcplan c,\n" + " setup_loc_ident lb,\n" + " section_inventory i\n" + " WHERE a.loc_ident = ?\n" + " AND a.pms_master_wp_type_id = 8\n" + " AND a.loc_ident = b.loc_ident\n" + " AND b.source_table = 'PMS_MASTER_WP'\n" + " AND c.pms_treatment_id = a.pms_treatment_id\n" + " AND lb.source_table = 'SECTION_INVENTORY'\n" + " AND b.route_id = lb.route_id\n" + " AND (b.lane_dir = lb.lane_dir OR b.lane_dir = 0 OR lb.lane_dir = 0\n" + ")\n" + " AND b.offset_from < lb.offset_to\n" + </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> " AND b.offset_to > lb.offset_from\n" + " AND i.loc_ident = lb.loc_ident\n" + " AND section_status_id = 1\n" + " AND lb.offset_from < lb.offset_to) a,\n" + " qg_owner_ref_year b\n" + " WHERE b.eff_fiscal_year = a.plan_year\n" + " AND b.owner_id = a.owner_id\n" + " AND b.activity_id = a.activity_id"); for(int dsRow=1; dsRow<=dsIn.rowCount(); dsRow++) { if(new Integer(1).equals(ds.getItemInteger(i,"A PPROVED")) && !(new Integer(1).equals(ds.getItemIntegerOrig inal(i,"APPROVED")))) { dsInsert.retrieve(dsIn.getItemInteger(ds Row,"loc_ident")); if(dsInsert.rowCount()>0 && !dsInsert.isItemNull(1,"PROJECT_ID")) { throw new TRDIException("this will create the work order."); } } else if(new Integer(0).equals(ds.getItemInteger(i,"A PPROVED")) && (new Integer(1).equals(ds.getItemIntegerOrig inal(i,"APPROVED")))) { throw new TRDIException("this will delete the work order!"); } } import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; import com.agileassetsinc.core.*; Integer max_accum_years=PerfMod.getMaxAcc umYears(); /* Data store of GHG Coefficients*/ DataStore dsIRIGHgFactors = dl.createRetrieveDataStore("SELECT a.ca_ghg_iri_factors_id,a.ca_ghg_iri_fac tors_name,a.ca_ghg_iri_factor,a.ca_ghg _constant_factor FROM setup_ca_ghg_iri_factors a order by a.ca_ghg_iri_factors_id"); /* DataStore to hold material and construction GHG factor*/ DataStore dsIRIMaterialFactors = dl.createRetrieveDataStore("SELECT a.</pre>
328	GHG Calculation	

ID	Groovy Script Name	Groovy Script Text
		<pre> pms_treatment_id, a.module_id, nvl(a.ca_ghg_material_coef,0) as ca_ghg_material_coef FROM ms_treatment a WHERE a.module_id = 6"); dsIRIMaterialFactors.buildIndex("idx", "PMS_TREATMENT_ID"); Double[] GHGVeh = new Double[5]; //Array to store "IRI" coefficients Double[] coef_iri = new Double[5]; //Array to store "Constant" coefficients Double[] coef_const = new Double[5]; Double[] Volume= new Double[5]; //Populate iri and constant coeffs for (int i = 1; i <= 5; i++) { coef_iri[i-1]= dsIRIGhgFactors.getItemNumber(i,"CA_ GHG_IRI_FACTOR"); coef_const[i-1] =dsIRIGhgFactors.getItemNumber(i,"CA_ GHG_CONSTANT_FACTOR"); } String altAddition = (alt_in==0?"": "_ALT_" +alt_in); Double GHGTotal = 0.0; String column_in="CA_IRI_AVG"; PerformanceIndex pi = PI.get(column_in); Integer treatment; //Double LaneMiles; int accumYears; int NumYears; Integer node_in; Integer rec_num; String mapName; Integer deter_type; Integer num_years; Double coef; Double value_in; Double[] detValues; int row; for (int i = 1; i <= ds_in.rowCount(); i++) { GHGTotal=0.0; treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID"+ altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; //Get traffic values 0 if null Volume[0] = ds_in.getItemNumber(i,"CA_TRAFFIC_ CAR_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_CAR_VOL") ; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> Volume[1] = ds_in.getItemNumber(i,"CA_TRAFFIC_2 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_2AXL_VOL"); Volume[2] = ds_in.getItemNumber(i,"CA_TRAFFIC_3 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_3AXL_VOL"); Volume[3] = ds_in.getItemNumber(i,"CA_TRAFFIC_4 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_4AXL_VOL"); Volume[4] = ds_in.getItemNumber(i,"CA_TRAFFIC_5 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_5AXL_VOL"); //LaneMiles = ds_in.getItemNumber(i,"LANE_MILES") ==null ? 0.0:ds_in.getItemNumber(i,"LANE_MILE S"); accumYears = 1; node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); rec_num = ds_in.getItemInteger(i,"ENGINE_REC_I D"); mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; deter_type = alt_in == 0 ? 4:(Integer) pi.getAddDeterParm("DET_TYPE" + mapName); num_years = deter_type==1 deter_type==4?0:(Integer) pi.getAddDeterParm("NUM_YEARS" + mapName); coef = deter_type==1 deter_type==4?1.0:(Double) pi.getAddDeterParm("X_COEF" + mapName); value_in = ds_in.getItemNumber(i,column_in+altAd dition); detValues=PerfMod.DeteriorateArr(colu mn_in,node_in,rec_num,value_in,deter_ type==1?null:ds_add_deter,max_accum _years,coef,num_years); detValues[0]=value_in; while (accumYears <max_accum_years && detValues[accumYears]!=null) { for (int j = 0; j <= 4; j++) { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> GHGVeh[j]=(coef_iri[j]*(100- 12*(detValues[accumYears] + detValues[accumYears- 1])/5)+coef_const[j])*Volume[j]; } GHGTotal += GHGVeh[0]+GHGVeh[1]+GHGVeh[2]+G HG Veh[3]+GHGVeh[4]; accumYears++; if(alt_in!=0 &&(treatment != null treatment != 0)) { Double slabCrackPct = ds_in.getItemInteger(i, "CA_APCS_JPC_CK3_PCT"+ altAddition); if(treatment == 205 /*Grind and replace slabs*/){ /*0.01589x0.75x12*5280*CK3 PCT *Lane Miles + 0.00329x0.03125x12x5280*LaneMile*/ GHGTotal+= (755.09*(100- slabCrackPct)/100+6.5142) } else if (treatment == 206 /*Replace slabs only*/){ /*0.01589x0.75x12*5280*CK3 PCT *Lane Miles */ GHGTotal+= 755.09*(100- slabCrackPct)/100 } else { row = dsIRIMaterialFactors.findByIndex("idx", treatment); GHGTotal += dsIRIMaterialFactors.getItemNumber(ro w,"CA_GHG_MATERIAL_COEF"); } /* Lane miles already part of analysis weight */ } } GHGTotal=-1*GHGTotal/accumYears; /*set to negative value because if used as an objective the software will always maximize these columns thus maximize the negative=minimize the positive*/ ds_in.setItemNumber(i,col_in+altAdditio n,TRDIUtils.roundNumber(GHGTotal,4)) ; ds_in.setItemNumber(i,col_in+"_DIFF",T RDIUtils.roundNumber(GHGTotal- ds_in.getItemNumber(i,col_in),2)); } </pre>

ID	Groovy Script Name	Groovy Script Text
347	Calculate Preservation Benefit	<pre> import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; Integer max_accum_years=PerfMod.getMaxAccumYears(); Double[] benWeight = new Double[4]; benWeight[0]=2.0; benWeight[1]=1.0; benWeight[2]=2.0; benWeight[3]=1.0; Double[] benThreshold = new Double[4]; benThreshold[0]=30.0; benThreshold[1]=2.0; benThreshold[2]=10.0; benThreshold[3]=0.25; String[] column_in = new String[4]; column_in[0]="CA_WPCV"; column_in[1]="CA_FLX_TCV"; column_in[2]="CA_APCS_JPC_CK3_PCT"; column_in[3]="CA_FHT_VAL"; String altAddition = (alt_in==0?"": "_ALT_" + alt_in); PerformanceIndex [] pi=new PerformanceIndex[4]; pi[0]=PI.get("CA_WPCV"); pi[1]=PI.get("CA_FLX_TCV"); pi[2]=PI.get("CA_APCS_JPC_CK3_PCT"); pi[3]=PI.get("CA_FHT_VAL"); for (int i = 1; i <= ds_in.rowCount(); i++) { Integer treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID" + altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; Double[][] ben = new Double[4][]; Double[] wben = new Double[4]; int wcid= ds_in.getItemInteger(i,"WC_ID")==1?0:2 ; for (int piArrayIndex=wcid;piArrayIndex<=wcid+ 1;piArrayIndex++) { int accumYears = 1; Integer node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); Integer rec_num = ds_in. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getItemInteger(i,"ENGINE_REC_ID"); String mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; Integer deter_type = alt_in == 0 ? 4:(Integer) pi[piArrayIndex].getAddDeterParm("DET _TYPE" + mapName); Integer num_years = deter_type==1 deter_type==4?0:(Integer) pi[piArrayIndex].getAddDeterParm("NU M_YEARS" + mapName); Double coef = deter_type==1 deter_type==4?1.0:(Double) pi[piArrayIndex].getAddDeterParm("X_C OEF" + mapName); Double value_in = ds_in.getItemNumber(i,column_in[piArra yIndex]+altAddition); Double[] detValues=PerfMod.DeteriorateArr(colu mn_in[piArrayIndex],node_in,rec_num,v alue_in,deter_type==1?null:ds_add_det er,max_accum_years,coef,num_years); // (2^x)^2 = 2^(2*x), so we don't need to square later. ben[piArrayIndex] = new Double[max_accum_years]; ben[piArrayIndex][0] = Math.pow(2,- 2*(pi[piArrayIndex].ConvertFrom100(val ue_in)/benThreshold[piArrayIndex])); while (accumYears < max_accum_years) { ben[piArrayIndex][accumYears] = detValues[accumYears]!=null ? Math.pow(2,- 2*(pi[piArrayIndex].ConvertFrom100(det Values[accumYears])/benThreshold[piAr rayIndex])) : 0; accumYears++; } wben[piArrayIndex] = Math.pow(benWeight[piArrayIndex],2) } Double totBenefit = 0.5*Math.sqrt(wben[wclد]*ben[wclد][0]+ wben[wclد+1]*ben[wclد+1][0]); for (int accumYears = 1;accumYears < max_accum_years;accumYears++) { totBenefit += Math.sqrt(wben[wclد]*ben[wclد][accumY ears]+wben[wclد+1]*ben[wclد+1][accum Years]); } // Assume the last trapezoid has </pre>

ID	Groovy Script Name	Groovy Script Text
		<p>zero area and don't subtract half...</p> <p>ds_in.setItemNumber(i,col_in+altAddition,TRDIUtils.roundNumber(totBenefit,2));</p>
343	Calculate Preservation Benefit	<pre> ds_in.setItemNumber(i,col_in+"_DiFF",TRDIUtils.roundNumber(totBenefit-ds_in.getItemNumber(i,col_in),2)); } import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; Integer max_accum_years=PerfMod.getMaxAccumYears(); double wpcv_wt=4.0; double flx_tcv_wt=1.0; double jcp_3CK_wt=4.0; double fht_wt=1.0; Double[] benThreshold = new Double[4]; benThreshold[0]=30.0; benThreshold[1]=2.0; benThreshold[2]=10.0; benThreshold[3]=0.25; /* * Original request fomula: * Case when a.wc_id=1 then * Benefit_pres=0.5*(wpcv_wt*(100-CA_WPCV) +flex_tci_wt * CA_FLEX_TCI) * When a.wc_id=2 then * Benefit_pres=0.5*greatest((jcp_3CK_wt* (100-6* CA_APCS_JPC_CK3_PCT) +(100*fht_wt*((5.8- CA_FHT_VAL)/5.8)),0) * Else Benefit_pres =0 * End * * Formula that gives the same value always but uses indices converted in 100 - 0 scale: * wc_id=1 * 0.5*(wpcv_wt* a.CA_WPCV +flex_tci_wt*a.CA_FLEX_TCI) * wc_id=2 * 0.5*greatest(jcp_3CK_wt*(100-6*(100 - CA_APCS_JPC_CK3_PCT)) +fht_wt* CA_FHT_VAL,0) */ String[] column_in = new String[4]; column_in[0]="CA_WPCV"; column_in[1]="CA_FLX_TCV"; column_in[2]="CA_APCS_JPC_CK3_P CT"; column_in[3]="CA_FHT_VAL"; String altAddition = (alt_in==0?"":") </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> _ALT_" + alt_in); PerformanceIndex [] pi=new PerformanceIndex[4]; pi[0]=PI.get("CA_WPCV"); pi[1]=PI.get("CA_FLX_TCV"); pi[2]=PI.get("CA_APCS_JPC_CK3_PCT"); pi[3]=PI.get("CA_FHT_VAL"); for (int i = 1; i <= ds_in.rowCount(); i++) { Integer treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID" + altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; Double[] ben = new Double[4]; ben[0]=0.0; ben[1]=0.0; ben[2]=0.0; ben[3]=0.0; int wclD=ds_in.getItemInteger(i,"WC_ID"); for (int piArrayIndex=(wclD==1?0:2);piArrayIndex<=(wclD==1?1:3);piArrayIndex++) { int accumYears = 1; Integer node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); Integer rec_num = ds_in.getItemInteger(i,"ENGINE_REC_ID"); String mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; Integer deter_type = alt_in == 0 ? 4:(Integer) pi[piArrayIndex].getAddDeterParm("DET _TYPE" + mapName); Integer num_years = deter_type==1 deter_type==4?0:(Integer) pi[piArrayIndex].getAddDeterParm("NUM_YEARS" + mapName); Double coef = deter_type==1 deter_type==4?1.0:(Double) pi[piArrayIndex].getAddDeterParm("X_COEF" + mapName); Double value_in = ds_in.getItemNumber(i,column_in[piArrayIndex]+altAddition); Double[] detValues=PerfMod.DeteriorateArr(colu </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> value_in,deter_type==1?null:ds_add_deter,max_accum_years,coef,num_years); detValues[0] = [benThreshold[piArrayIndex]- pi[piArrayIndex].ConvertFrom100(value_in),0].max(); while (accumYears < max_accum_years && detValues[accumYears]!=null) { detValues[accumYears] = [benThreshold[piArrayIndex]- pi[piArrayIndex].ConvertFrom100(detValues[accumYears]),0].max(); ben[piArrayIndex] += 0.5*(detValues[accumYears] + detValues[accumYears -1]); accumYears++; } } Double totBenefit= (wcld==1 ? (wpcv_wt*ben[0]/benThreshold[0]+flx_tcv_wt*ben[1]/benThreshold[1])/(wpcv_wt+flx_tcv_wt) : (jcp_3CK_wt*ben[2]/benThreshold[2]+fht_wt*ben[3]/benThreshold[3])/(jcp_3CK_wt+fht_wt)); ds_in.setItemNumber(i,col_in+altAddition,TRDIUtils.roundNumber(totBenefit,2)); </pre>
341	Capture Conditions Prior to	<pre> ds_in.setItemNumber(i,col_in+"_DIFF",TRDIUtils.roundNumber(totBenefit-ds_in.getItemNumber(i,col_in),2)); //This script saves the current pre-treated condition into a separate column //Allows for later comparison of how much improvement was done between pre and post treatment import com.agileassetsinc.core.*; import com.agileassetsinc.main.optim.*; Double convertVal; if (alt_in==0) { DataStore dsPi=dl.createRetrieveDataStore("select COLUMN_ID, MODEL_START, MODEL_FINISH from PMS_MODEL_ATTR"); dsPi.buildIndex("idx","COLUMN_ID"); int rowCK3 = dsPi.findByIndex("idx","CA_APCS_JPC_CK3_PCT"); int rowFHT = dsPi.findByIndex("idx","CA_FHT_VAL"); int rowTCV = dsPi.findByIndex("idx","CA_FLX_TCV"); int rowLRI = dsPi.findByIndex </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ("idx", "CA_IRI_AVG"); int rowMPD = dsPi.findByIndex("idx", "CA_MPD_MM"); int rowWPC = dsPi.findByIndex("idx", "CA_WPCV"); for (int i = 1; i <= ds_in.rowCount(); i++) { if (ds_in.getItemNumber(i,"CA_APCS_JP C_CK3_PCT")!=null) { convertVal=convertPI.toRealScale(dsPi. getItemNumber(rowCK3, "MODEL_START"),dsPi.getItemNumber (rowCK3, "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_APCS_JPC _CK3_PCT")); ds_in.setItemNumber(i,"CA_APCS_JPC _CK3_PCT_PR",convertVal); } if (ds_in.getItemNumber(i,"CA_FHT_VAL")!=null) { convertVal=convertPI.toRealScale(dsPi. getItemNumber(rowFHT, "MODEL_START"),dsPi.getItemNumber (rowFHT, "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_FHT_VAL")); ds_in.setItemNumber(i,"CA_FHT_VAL_ PR",convertVal); } if (ds_in.getItemNumber(i,"CA_FLX_TCV")!=null) { convertVal=convertPI.toRealScale(dsPi. getItemNumber(rowTCV, "MODEL_START"),dsPi.getItemNumber (rowTCV, "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_FLX_TCV")); ds_in.setItemNumber(i,"CA_FLX_TCV_ PR",convertVal); } if (ds_in.getItemNumber(i,"CA_IRI_AVG")!= null) { convertVal=convertPI.toRealScale(dsPi. getItemNumber(rowIRI, "MODEL_START"),dsPi.getItemNumber (rowIRI, "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_IRI_AVG")); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_MPD_MM")); ds_in.setItemNumber(i,"CA_MPD_MM_ PR",convertVal); } if (ds_in.getItemNumber(i,"CA_WPCV")!= null) { convertVal=convertPI.toRealScale(dsPi. getItemNumber(rowWPC, "MODEL_START"),dsPi.getItemNumber (rowWPC, "MODEL_FINISH"), ds_in.getItemNumber(i,"CA_WPCV")); ds_in.setItemNumber(i,"CA_WPCV_PR" ,convertVal); } } } class convertPI { static double toRealScale(double startValue, double endValue, double in_value) { int dir = startValue>endValue?-1:1; double alpha = 100/Math.abs(startValue - endValue) ; double shift = Math.min(startValue, endValue) ; return Math.min(Math.max(startValue, endValue),Math.max(Math.min(startValu e, endValue), shift + dir*(100 * (1 + dir)/2 import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; import com.agileassetsinc.core.*; //General Vars Integer max_accum_years=PerfMod.getMaxAcc umYears(); String altAddition = (alt_in==0?"": "_ALT_" +alt_in); Integer treatment; int accumYears; Integer node_in; Integer rec_num; String mapName; Integer deter_type; Integer num_years; Double coef; Double value_in; Double[] detValues; // Get Variables for Preservation Benefit double wpcv_wt=1.0; </pre>
337	Combined Caltrans Benefit Assign to	

ID	Groovy Script Name	Groovy Script Text
		<pre> double flx_tcv_wt=1.0; double jcp_3CK_wt=1.0; double fht_wt=1.0; String[] columns_in = new String[4]; columns_in[0]="CA_WPCV"; columns_in[1]="CA_FLX_TCV"; columns_in[2]="CA_APCS_JPC_CK3_P CT"; columns_in[3]="CA_FHT_VAL"; PerformanceIndex [] pi_arr=new PerformanceIndex[4]; pi_arr[0]=PI.get("CA_WPCV"); pi_arr[1]=PI.get("CA_FLX_TCV"); pi_arr[2]=PI.get("CA_APCS_JPC_CK3_ PCT"); pi_arr[3]=PI.get("CA_FHT_VAL"); Double[] ben_arr = new Double[4]; // Variables for Road User Costs /* Data store of Road User Cost Coefficients*/ DataStore dsVehicleTypesCosts = dl.createRetrieveDataStore("select A.CA_RUC_VEH_TYPE_ID as veh_type, A.CA_RUC_A_COEF as a_coef, A.CA_RUC_B_COEF as b_coef from SETUP_CA_RUC_VEH_TYPE a order by A.CA_RUC_VEH_TYPE_ID"); //Array to store "a" coefficients Double[] coef_a = new Double[5]; //Array to store "b" coefficients Double[] coef_b = new Double[5]; //Array to store calculated user costs as function of IRI Double[] RUC = new Double[5]; //Populate a and b coeffs for (int i = 1; i <= 5; i++) { coef_a[i-1]= dsVehicleTypesCosts.getItemNumber(i," A_COEF"); coef_b[i-1] =dsVehicleTypesCosts.getItemNumber(i ,"B_COEF"); } Double adjLaneADT; Double Vol2axl; Double Vol3axl; Double Vol4axl; Double Vol5axl; Double benUC; //Variables to store untransformed and transformed IRI values reallIRI= normal IRI, indIRI is 0 to 100 based on range of 20 to 500 Double reallIRI; Double indIRI; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> Double ben = 0.0; String column_in="CA_IRI_AVG"; PerformanceIndex pi = PI.get(column_in); //Weighting factors for mixed Preservation and IRI Traffic benefit number Double benIRITrafficWt=1.0; Double benPreservationWt=1.0; // Variables to Store Benefit values Double totBenefitPres; Double benIRITraffic=0.0; Double totBenefit=0.0; //Loop through the Optimization engine for (int i = 1; i <= ds_in.rowCount(); i++) { //get needed column values //Get traffic values 0 if null adjLaneADT = ds_in.getItemNumber(i,"CA_EQUIV_AD J_ADT_LANE")==null ? 0.0:ds_in.getItemNumber(i,"CA_EQUIV_ ADJ_ADT_LANE") ; Vol2axl = ds_in.getItemNumber(i,"CA_TRAFFIC_2 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_2AXL_VOL"); Vol3axl = ds_in.getItemNumber(i,"CA_TRAFFIC_3 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_3AXL_VOL"); Vol4axl = ds_in.getItemNumber(i,"CA_TRAFFIC_4 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_4AXL_VOL"); Vol5axl = ds_in.getItemNumber(i,"CA_TRAFFIC_5 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_5AXL_VOL"); //get treatment and WC ID treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID"+ altAddition); int wcId=ds_in.getItemInteger(i,"WC_ID"); //Initialize benefit arrays for PReservation benefits ben_arr[0]=0.0; ben_arr[1]=0.0; ben_arr[2]=0.0; ben_arr[3]=0.0; //Get basic Modeling parameters node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; //Chek if treatment is null or do nothing if so skip to next if (alt_in>0 &&(treatment == null treatment == 0)) continue; //Calculate CALTRANS Preservation Benefit /* * Original request fomula: * Case when a.wc_id=1 then * Benefit_pres=0.5*(wpcv_wt*(100- CA_WPCV) +flex_tci_wt * CA_FLEX_TCI) * When a.wc_id=2 then * Benefit_pres=0.5*greatest((jcp_3CK_wt* (100-6* CA_APCS_JPC_CK3_PCT) +(100*fht_wt*((5.8- CA_FHT_VAL)/5.8)),0) * Else Benefit_pres =0 * End * * Formula that gives the same value always but uses indices converted in 100 - 0 scale: * wc_id=1 * 0.5*(wpcv_wt* a.CA_WPCV +flex_tci_wt*a.CA_FLEX_TCI) * wc_id=2 * 0.5*greatest(jcp_3CK_wt*(100-6*(100 - CA_APCS_JPC_CK3_PCT)) +fht_wt* CA_FHT_VAL,0) */ //Loop to get area values for indexes in preservation benefit for (int piArrayIndex=(wcId==1?0:2);piArrayInde x<=(wcId==1?1:3);piArrayIndex++) { accumYears = 1; deter_type = alt_in == 0 ? 4:(Integer) pi_arr[piArrayIndex].getAddDeterParm(" DET_TYPE" + mapName); num_years = deter_type==1 deter_type==4?0:(Integer) pi_arr[piArrayIndex].getAddDeterParm(" NUM_YEARS" + mapName); coef = deter_type==1 deter_type==4?1.0:(Double) pi_arr[piArrayIndex].getAddDeterParm(" X_COEF" + mapName); value_in = ds_in.getItemNumber(i,columns_in[piArr ayIndex]+altAddition); detValues=PerfMod.DeteriorateArr(colu mns_in[piArrayIndex],node_in,rec_num, value_in,deter_type==1?null:ds_add_det er,max_accum_years, </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> coef,num_years); detValues[0]=value_in; while (accumYears <max_accum_years && detValues[accumYears]!=null) { ben_arr[piArrayIndex] += 0.5*(detValues[accumYears] + detValues[accumYears -1]) ; accumYears++; } } //Calculate Preservation Benefit (also used later) totBenefitPres= (wclId==1 ? 0.5*(wpcv_wt* ben_arr[0] +flx_tcv_wt*ben_arr[1]) : 0.5* (jcp_3CK_wt*(100-6*(100 - ben_arr[2])) +fht_wt* ben_arr[3])); ds_in.setItemNumber(i,col_in+altAdditio n,TRDIUtils.roundNumber(totBenefitPre s,2)); //Calculate IRI Road User Cost benefit and IRI Traffic Weighted Benefit /* 1000-(((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) – 1.5 * ((2_Axle_volume*LDF_i) + (3_Axle_volume*LDF_i) + (4_Axle_volume*LDF_i) + (5_Axle_volume*LDF_i))) * Cars_RUC + (2_Axle_volume*LDF_i) * 2_axle_RUC + (3_Axle_volume*LDF_i)* 3_axle_RUC + (4_Axle_volume*LDF_i) * 4_axle_RUC + (5_Axle_volume*LDF_i) * 5_axle_RUC)/10^3 */ //Initialize User Cost Benefit and IRI Benefit benUC=0.0; ben=0.0; //reset years back to one accumYears = 1; //Get deterioration information for IRI Performance index deter_type = alt_in == 0 ? 4:(Integer) pi.getAddDeterParm("DET_TYPE" + mapName); num_years = deter_type==1 deter_type==4?0:(Integer) pi.getAddDeterParm("NUM_YEARS" + mapName); coef = deter_type==1 deter_type==4?1.0:(Double) pi. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getAddDeterParm("X_COEF" + mapName); //Get Current IRI value_in = ds_in.getItemNumber(i,column_in+altAd dition); //Deteriorate IRI detValues=PerfMod.DeteriorateArr(colu mn_in,node_in,rec_num,value_in,deter_ type==1?null:ds_add_deter,max_accum _years,coef,num_years); detValues[0]=value_in; while (accumYears <max_accum_years && detValues[accumYears]!=null) { indIRI=0.5*(detValues[accumYears] + detValues[accumYears -1]); realIRI=(4.8*(100- (indIRI))+20); for (int j = 0; j <= 4; j++) { /* Take Trapezoid approximation of IRI for Regular Traffic weighted IRI for User Cost one convert DET values from 100-0 back natural IRI ((100- IRI100)*480/100)+20) multiply predicted IRI value against RUC formula RUC=a*exp(b*IRI) per class of vehicle */ RUC[j]=coef_a[j]*Math.exp(coef_b[j]*real IRI); } ben += indIRI; benUC += 1000-((adjLaneADT - 1.5 * (Vol2axl + Vol3axl + Vol4axl + Vol5axl))*RUC[0] + Vol2axl * RUC[1] + Vol3axl*RUC[2] + Vol4axl * RUC[3] + Vol5axl * RUC[4])/1000; accumYears++; } //Calculate Traffic Weighted IRI Benefit /* Original formula Ben_IRI_Traffic_Weighted = (IRI_average_max – IRI_average) * ((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) / 10^6 ((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) / 10^6 is precomputed in network master */ ds_in.setItemNumber(i,"CA_BEN_IRI_R UC"+altAddition,TRDIUtils.roundNumber (benUC,4)); benIRITraffic=ben * ds_in.getItemNumber(i,"CA_BEN_TRAF FIC_WEIGHT"); ds_in.setItemNumber(i,"CA_BEN_IRI_T </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (benIRITraffic,4)); // Calculate Benefit IRI Traffic Preservation combined totBenefit= (benIRITraffic==null?0:benIRITraffic)*benIRITrafficWt+(totBenefitPres==null?0:totBenefitPres)*benPreservationWt; </pre>
315	Benefit IRI Traffic Preservation	<pre> ds_in.setItemNumber(i,"CA_BEN_IRI_T RAF_PRES"+altAddition,TRDIUtils.round dNumber(totBenefit,4)); } import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; String altAddition = (alt_in==0?"":"_ALT_"+alt_in); Double benIRITrafficWt=1.0; Double benPreservationWt=1.0; Integer treatment; Double totBenefit; Double benIRITraffic=0.0; Double benPreservation=0.0; for (int i = 1; i <= ds_in.rowCount(); i++) { treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID"+ altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; benIRITraffic=ds_in.getItemNumber(i,"C A_BEN_IRI_TRAFFIC"+altAddition); benPreservation=ds_in.getItemNumber(i ,"CA_BENEFIT_PRES"+altAddition); totBenefit= (benIRITraffic==null?0:benIRITraffic)*be nIRITrafficWt+(benPreservation==null?0 :benPreservation)*benPreservationWt; ds_in.setItemNumber(i,col_in+altAdditio n,TRDIUtils.roundNumber(totBenefit,4)); ds_in.setItemNumber(i,col_in+"_DIFF",T RDIUtils.roundNumber(totBenefit- ds_in.getItemNumber(i,col_in),2)); } </pre>
302	Calculate CALTRANS Preservation	<pre> import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> Integer max_accum_years=PerfMod.getMaxAccumYears(); double wpcv_wt=1.0; double flx_tcv_wt=1.0; double jcp_3CK_wt=1.0; double fht_wt=1.0; /* * Original request fomula: * Case when a.wc_id=1 then * Benefit_pres=0.5*(wpcv_wt*(100-CA_WPCV) +flex_tci_wt *CA_FLEX_TCI) * When a.wc_id=2 then * Benefit_pres=0.5*greatest((jcp_3CK_wt*(100-6* CA_APCS_JPC_CK3_PCT)+(100*fht_wt*((5.8-CA_FHT_VAL)/5.8)),0) * Else Benefit_pres =0 * End * * Formula that gives the same value always but uses indices converted in 100 - 0 scale: * wc_id=1 * 0.5*(wpcv_wt* a.CA_WPCV +flex_tci_wt*a.CA_FLEX_TCI) * wc_id=2 * 0.5*greatest(jcp_3CK_wt*(100-6*(100 - CA_APCS_JPC_CK3_PCT)) +fht_wt*CA_FHT_VAL,0) */ String[] column_in = new String[4]; column_in[0]="CA_WPCV"; column_in[1]="CA_FLX_TCV"; column_in[2]="CA_APCS_JPC_CK3_PCT"; column_in[3]="CA_FHT_VAL"; String altAddition = (alt_in==0?"": "_ALT_" +alt_in); PerformanceIndex [] pi=new PerformanceIndex[4]; pi[0]=PI.get("CA_WPCV"); pi[1]=PI.get("CA_FLX_TCV"); pi[2]=PI.get("CA_APCS_JPC_CK3_PCT"); pi[3]=PI.get("CA_FHT_VAL"); for (int i = 1; i <= ds_in.rowCount(); i++) { Integer treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID"+ altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> continue; Double[] ben = new Double[4]; ben[0]=0.0; ben[1]=0.0; ben[2]=0.0; ben[3]=0.0; int wclD=ds_in.getItemInteger(i,"WC_ID"); for (int piArrayIndex=(wclD==1?0:2);piArrayIndex<=(wclD==1?1:3);piArrayIndex++) { int accumYears = 1; Integer node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); Integer rec_num = ds_in.getItemInteger(i,"ENGINE_REC_ID"); String mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; Integer deter_type = alt_in == 0 ? 4:(Integer) pi[piArrayIndex].getAddDeterParm("DET _TYPE" + mapName); Integer num_years = deter_type==1 deter_type==4?0:(Integer) pi[piArrayIndex].getAddDeterParm("NUM _YEARS" + mapName); Double coef = deter_type==1 deter_type==4?1.0:(Double) pi[piArrayIndex].getAddDeterParm("X_COEF" + mapName); Double value_in = ds_in.getItemNumber(i,column_in[piArrayIndex]+altAddition); Double[] detValues=PerfMod.DeteriorateArr(column_in[piArrayIndex],node_in,rec_num,value_in,deter_type==1?null:ds_add_deter,max_accum_years,coef,num_years); detValues[0]=value_in; while (accumYears <max_accum_years && detValues[accumYears]!=null) { ben[piArrayIndex] += 0.5*(detValues[accumYears] + detValues[accumYears -1]) ; accumYears++; } } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> Double totBenefit= (wclD==1 ? 0.5*(wpcv_wt* ben[0] +flx_tcv_wt*ben[1]) : 0.5* (jcp_3CK_wt*(100-6*(100 - ben[2])) +fht_wt* ben[3])); /* XXX: Jeremy Lea 20140523 suggested change. Double totBenefit= (wclD==1 ? (wpcv_wt*ben[0]+flx_tcv_wt*ben[1])/(wp cv_wt+flx_tcv_wt) : (jcp_3CK_wt*ben[2]+fht_wt*ben[3])/(jcp_ 3CK_wt+fht_wt)); */ </pre>
312	Calculate Traffic Weighted IRI Benefit	<pre> ds_in.setItemNumber(i,col_in+altAdditio import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; /* Original formula Ben_IRI_Traffic_Weighted = (IRI_average_max – IRI_average) * ((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) / 10^6 ((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) / 10^6 is precomputed in network master */ Integer max_accum_years=PerfMod.getMaxAcc umYears(); String altAddition = (alt_in==0?"": "_ALT_" +alt_in); Double ben = 0.0; String column_in="CA_IRI_AVG"; PerformanceIndex pi = PI.get(column_in); for (int i = 1; i <= ds_in.rowCount(); i++) { ben=0.0; Integer treatment = ds_in.getItemInteger(i, "PMS_TREATMENT_ID" + altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; int accumYears = 1; Integer node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); </pre>

ID	Groovy Script Name	Groovy Script Text
313	IRI Road User Cost benefit	<pre> Integer rec_num = ds_in.getItemInteger(i,"ENGINE_REC_I D"); String mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; Integer deter_type = alt_in == 0 ? 4:(Integer) pi.getAddDeterParm("DET_TYPE" + mapName); Integer num_years = deter_type==1 deter_type==4?0:(Integer) pi.getAddDeterParm("NUM_YEARS" + mapName); Double coef = deter_type==1 deter_type==4?1.0:(Double) pi.getAddDeterParm("X_COEF" + mapName); Double value_in = ds_in.getItemNumber(i,column_in+altAd dition); Double[] detValues=PerfMod.DeteriorateArr(colu mn_in,node_in,rec_num,value_in,deter_ type==1?null:ds_add_deter,max_accum _years,coef,num_years); detValues[0]=value_in; while (accumYears <max_accum_years && detValues[accumYears]!=null) { ben += 0.5*(detValues[accumYears] + detValues[accumYears -1]) ; accumYears++; } Double totBenefit=ben * ds_in.getItemNumber(i,"CA_BEN_TRAF FIC_WEIGHT"); ds_in.setItemNumber(i,col_in+altAdditio n,TRDIUtils.roundNumber(totBenefit,2)); ds_in.setItemNumber(i,col_in+"_DIFF",T RDIUtils.roundNumber(totBenefit- import com.agileassetsinc.main.optim.*; import com.agileassetsinc.core.TRDIUtils; import com.agileassetsinc.core.*; /* 1000-(((Adjusted_ADT + 0.5 Truck_volume) / No_of_lanes) - 1.5 * ((2_Axle_volume*LDF_i) + (3_Axle_volume*LDF_i) + (4_Axle_volume*LDF_i) + (5_Axle_volume*LDF_i)))*) Cars_RUC + (2_Axle_volume*LDF_i) * 2_axle_RUC + </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (3_Axle_volume*LDF_i)* 3_axle_RUC + (4_Axle_volume*LDF_i) * 4_axle_RUC + (5_Axle_volume*LDF_i) * 5_axle_RUC)/10^3 */ Integer max_accum_years=PerfMod.getMaxAccumYears(); /* Data store of Road User Cost Coefficients*/ DataStore dsVehicleTypesCosts = dl.createRetrieveDataStore("select A.CA_RUC_VEH_TYPE_ID as veh_type, A.CA_RUC_A_COEF as a_coef, A.CA_RUC_B_COEF as b_coef from SETUP_CA_RUC_VEH_TYPE a order by A.CA_RUC_VEH_TYPE_ID"); //Array to store "a" coefficients Double[] coef_a = new Double[5]; //Array to store "b" coefficients Double[] coef_b = new Double[5]; //Array to store calculated user costs as function of IRI Double[] RUC = new Double[5]; //Populate a and b coeffs for (int i = 1; i <= 5; i++) { coef_a[i-1]= dsVehicleTypesCosts.getItemNumber(i," A_COEF"); coef_b[i-1] =dsVehicleTypesCosts.getItemNumber(i ,"B_COEF"); } String altAddition = (alt_in==0?"": "_ALT_" +alt_in); Double ben = 0.0; String column_in="CA_IRI_AVG"; PerformanceIndex pi = PI.get(column_in); Integer treatment; Double adjLaneADT; Double Vol2axl; Double Vol3axl; Double Vol4axl; Double Vol5axl; int accumYears; Integer node_in; Integer rec_num; String mapName; Integer deter_type; Integer num_years; Double coef; Double value_in; Double[] detValues; Double indIRI; Double reallIRI; for (int i = 1; i <= ds_in.rowCount(); i++) { ben=0.0; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> "PMS_TREATMENT_ID"+ altAddition); if (alt_in>0 &&(treatment == null treatment == 0)) continue; //Get traffic values 0 if null adjLaneADT = ds_in.getItemNumber(i,"CA_EQUIV_AD J_ADT_LANE")==null ? 0.0:ds_in.getItemNumber(i,"CA_EQUIV_ ADJ_ADT_LANE") ; Vol2axl = ds_in.getItemNumber(i,"CA_TRAFFIC_2 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_2AXL_VOL"); Vol3axl = ds_in.getItemNumber(i,"CA_TRAFFIC_3 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_3AXL_VOL"); Vol4axl = ds_in.getItemNumber(i,"CA_TRAFFIC_4 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_4AXL_VOL"); Vol5axl = ds_in.getItemNumber(i,"CA_TRAFFIC_5 AXL_VOL")==null ? 0.0:ds_in.getItemNumber(i,"CA_TRAFFI C_5AXL_VOL"); accumYears = 1; node_in = ds_in.getItemInteger(i,"MOD_NODE_ID" +altAddition); rec_num = ds_in.getItemInteger(i,"ENGINE_REC_I D"); mapName = ds_in.getItemInteger(i, "ENGINE_REC_ID") + (alt_in == 0 ? "A":alt_in == 1 ? "X" : alt_in == 2 ? "Y" : "Z") + treatment; deter_type = alt_in == 0 ? 4:(Integer) pi.getAddDeterParm("DET_TYPE" + mapName); num_years = deter_type==1 deter_type==4?0:(Integer) pi.getAddDeterParm("NUM_YEARS" + mapName); coef = deter_type==1 deter_type==4?1.0:(Double) pi.getAddDeterParm("X_COEF" + mapName); value_in = ds_in.getItemNumber(i,column_in+altAd dition); detValues=PerfMod.DeteriorateArr(colu mn_in,node_in,rec_num,value_in,deter_ type==1?null:ds_add_deter,max_accum </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> realIRI=(4.8*(100- (indIRI))+20); for (int j = 0; j <= 4; j++) { /* Take Trapezoid approximation of IRI for Regular Traffic weighted IRI for User Cost one convert DET values from 100-0 back natural IRI ((100- IRI100)*480/100)+20) multiply predicted IRI value against RUC formula RUC=a*exp(b*IRI) per class of vehicle */ RUC[j]=coef_a[j]*Math.exp(coef_b[j]*real IRI); } ben += 1000-((adjLaneADT - 1.5 * (Vol2axl + Vol3axl + Vol4axl + Vol5axl))*RUC[0] + Vol2axl * RUC[1] + Vol3axl*RUC[2] + Vol4axl * RUC[3] + Vol5axl * RUC[4])/1000; accumYears++; } ds_in.setItemNumber(i,col_in+altAdditio n,TRDIUtils.roundNumber(ben,4)); </pre>
-1040	General: Sat-Sun 7.5Hrs	<pre> ds_in.setItemNumber(i,col_in+"_DIFF",T RDIUtils.roundNumber(ben- ds_in.getItemNumber(i,col_in),2)); import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 0){ eff_hours = 0; } else{ eff_hours = 7.5; } return eff_hours; </pre>
-1024	General: Tue - Thurs 9.5 hrs (Fri 9)	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 2 dayOfWeek == 7){ eff_hours = 0; } else if(dayOfWeek == 6){ eff_hours = 9.0; } else{ eff_hours = 9.5; } </pre>

ID	Groovy Script Name	Groovy Script Text
		return eff_hours;
-1023	General: Mon - Thurs 9.5 hrs (Mon	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 6 dayOfWeek == 7){ eff_hours = 0; } else if(dayOfWeek == 2){ eff_hours = 9.0; } else{ eff_hours = 9.5; } return eff_hours; </pre>
-1022	General: Mon - Thurs 9.5 hrs (Thurs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 6 dayOfWeek == 7){ eff_hours = 0; } else if(dayOfWeek == 5){ eff_hours = 9.0; } else{ eff_hours = 9.5; } return eff_hours; </pre>
-1021	General: AWS4	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int DayOfWeek = cal.get(Calendar.DAY_OF_WEEK); int WeekOfYear= cal.get(Calendar.WEEK_OF_YEAR); double eff_hours; if((WeekOfYear%2 == 1) && (DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8.75; } else if((WeekOfYear%2 == 0) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8; } </pre>

ID	Groovy Script Name	Groovy Script Text
-1020	General: AWS3	<pre> else{ eff_hours = 0; } return eff_hours; import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int DayOfWeek = cal.get(Calendar.DAY_OF_WEEK); int WeekOfYear= cal.get(Calendar.WEEK_OF_YEAR); double eff_hours; if((WeekOfYear%2 == 1) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8; } else if((WeekOfYear%2 == 0) && (DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8.75; } else{ eff_hours = 0; } return eff_hours; </pre>
-1007	General: AWS2	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int DayOfWeek = cal.get(Calendar.DAY_OF_WEEK); int WeekOfYear= cal.get(Calendar.WEEK_OF_YEAR); double eff_hours; if((WeekOfYear%2 == 1) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5)){ eff_hours = 8.75; } else if((WeekOfYear%2 == 0) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8; } else{ eff_hours = 0; } return eff_hours; </pre>
-1006	General: AWS1	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int DayOfWeek = cal.get(Calendar.DAY_OF_WEEK); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> int WeekOfYear= cal.get(Calendar.WEEK_OF_YEAR); double eff_hours; if((WeekOfYear%2 == 1) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5 DayOfWeek == 6)){ eff_hours = 8; } else if((WeekOfYear%2 == 0) && (DayOfWeek == 2 DayOfWeek == 3 DayOfWeek == 4 DayOfWeek == 5)){ eff_hours = 8.75; } else{ eff_hours = 0; } return eff_hours; </pre>
-1013	General: Night Mon-Fri 9.5-9Hrs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 3 dayOfWeek == 4 dayOfWeek == 5){ eff_hours = 9.5; } else if(dayOfWeek == 2){ eff_hours = 4.0; } else if(dayOfWeek == 6){ eff_hours = 5.0; } else{ eff_hours = 0; } return eff_hours; </pre>
-1012	General: Sun-Sat 16.5Hrs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 7){ eff_hours = 24; } else{ eff_hours = 16.5; } return eff_hours; </pre>
-1011	General: Sun-Sat 24Hrs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 0){ eff_hours = 0; } </pre>
-1010	General: Sun-Sat 8Hrs	<pre> } else{ eff_hours = 24; } return eff_hours; import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 0){ eff_hours = 0; } </pre>
-1009	General: On call Thurs PM to Mon	<pre> } else{ eff_hours = 8; } return eff_hours; import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 3 dayOfWeek == 4){ eff_hours = 0; } else if(dayOfWeek == 2 dayOfWeek == 5){ eff_hours = 6.0; } else{ eff_hours = 24; } return eff_hours; </pre>
-1008	General: On call Fri PM to Mon AM	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek= cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 3 dayOfWeek == 4 dayOfWeek == 5){ eff_hours = 0; } else if(dayOfWeek == 2 dayOfWeek == 6){ eff_hours = 6.0; } else{ eff_hours = 24; } </pre>

ID	Groovy Script Name	Groovy Script Text
		return eff_hours;
-1004	General: 7 days 8hrs 3 days free	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfMonth= cal.get(Calendar.DAY_OF_MONTH); double eff_hours; if(dayOfMonth == 8 dayOfMonth == 9 dayOfMonth == 10 dayOfMonth == 18 dayOfMonth == 19 dayOfMonth == 20 dayOfMonth == 28 dayOfMonth == 29 dayOfMonth == 30){ eff_hours = 0; } else{ eff_hours = 8; } return eff_hours; </pre>
-1003	General: 4-10 Schedule Mon off	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 2 dayOfWeek == 7){ eff_hours = 0; } else{ eff_hours = 10; } return eff_hours; </pre>
-1002	General: 4-10 Schedule Fri off	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 6 dayOfWeek == 7){ eff_hours = 0; } else{ eff_hours = 10; } return eff_hours; </pre>
-1001	General: Regular - Mon - Fri 7.5 hrs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK); double eff_hours; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dayOfWeek == 1 dayOfWeek == 7){ eff_hours = 0; } else{ eff_hours = 7.5; } return eff_hours; </pre>
-1000	General: Regular - Mon - Fri 8 hrs	<pre> import java.util.Calendar; Calendar cal = new GregorianCalendar(yearIn, monthIn, dayIn); int dayOfWeek = cal.get(Calendar.DAY_OF_WEEK); double eff_hours; if(dayOfWeek == 1 dayOfWeek == 7){ eff_hours = 0; } else{ eff_hours = 8; } return eff_hours; </pre>
44	Age+1	<pre> Double newValueOut; if (valueIn==null) newValueOut=null; else newValueOut=valueIn + 1; return newValueOut; </pre>
335	Update CA_FLEX_TCI During	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 1) newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_FLX_TCV")/2.5),2))); else newValueOut=null; </pre>
305	Update CA_CK3_IND During Analysis	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 2) newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_APCS_JPC_CK3_PCT")/10),0.7))); else newValueOut=null; return newValueOut; </pre>
311	Update CA_IRI_IND During Analysis	<pre> Double newValueOut; newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_IRI_AVG")/275),3))); return newValueOut; </pre>
310	Update CA_MPD_IND During	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 1) newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_MPD_MM")/1500),2))); else newValueOut=null; </pre>

ID	Groovy Script Name	Groovy Script Text
309	Update CA_RUT_IND During Analysis	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 1) newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_RUT_AVG")/38),1.35))); else newValueOut=null; </pre>
308	Update CA_WPC_IND During	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 1) newValueOut= (100*Math.pow(2,- 1*Math.pow(((100- dsIn.getItemNumber(rowIn,"CA_WPCV"))/30),1.3))); else newValueOut=null; </pre>
307	Update CA_FHT_IND During Analysis	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"WC_ID")== 2) newValueOut= (100*Math.pow(2,- 1*Math.pow((dsIn.getItemNumber(rowIn, "CA_FHT_VAL")/0.15),1))); else newValueOut=null; </pre>
-20	General: Age + 1	<pre> Double newValueOut; if (valueIn==null) newValueOut=null; else newValueOut=valueIn + 1; return newValueOut; </pre>
163	Recaculate Deficient	<pre> import com.agileassetsinc.core.*; Double newValueOut; Double ci; Double ri; Integer func_class; Integer wc; func_class= dsIn.getItemNumber(rowIn,"ID_FUNC_C LASS_ID"); ci=dsIn.getItemNumber(rowIn,"ID_ASC_ CRACK_IND"); ri=dsIn.getItemNumber(rowIn,"ID_RIDE _IDX"); wc=dsIn.getItemNumber(rowIn,"WC_ID"); rut=dsIn.getItemNumber(rowIn,"ID_RUT _A_E"); newValueOut=1; if(! (wc==1 wc==2 wc==3)) { newValueOut =0; } else { if(func_class==1 func_class== 2 func_class==3 func_class== 8 func_class== 9 func_class== 10 func_class== 11) { if((ci < 50 ri < 2.5 rut>0.5)){ </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } else { newValueOut =0; } } else { if((ci < 40 ri < 2.0 rut>1.0)){ newValueOut =1; } else { newValueOut =0; } } } } </pre>
74	STRUCT_DEF	<pre> return newValueOut; Double newValueOut; if (dsIn.getItemNumber(rowIn,"PIN_CULV ERT_RATING")!=null && dsIn.getItemNumber(rowIn,"PIN_DECK_ RATING ")!=null) { if (dsIn.getItemNumber(rowIn,"PIN_CULV ERT_RATING")<=4 && dsIn.getItemNumber(rowIn,"PIN_DECK_ RATING ")<=4) newValueOut = 1 else 0} else 0; </pre>
66	MIN(CI,RI)	<pre> Double newValueOut; if (dsIn.getItemNumber(rowIn,"CI_100")!= null && dsIn.getItemNumber(rowIn,"RI_100")!=n ull) newValueOut=(dsIn.getItemNumber(row In,"CI_100")> dsIn.getItemNumber(rowIn,"RI_100")?0. 05*dsIn.getItemNumber(rowIn,"RI_100") : 0.05*dsIn.getItemNumber(rowIn,"CI_100 ")); </pre>
58	Improve to 80 (4)	<pre> Double ret; ret = 4.0; return ret; </pre>
345	Improve based on replaced slabs	<pre> Double ret; Double ck3; ck3=ds_in.getItemNumber(row_in,"CA_ APCS_JPC_CK3_PCT"); ret=val_in-0.6*(100-ck3); // 60(in/mi) / 100% 3rd stage. ret=ret<60?60:ret; return ret; </pre>
334	Improve iRI by 1pt	<pre> Double ret; ret=val_in-1; return ret; </pre>
287	Improve by - 60 (min 60)	<pre> Double ret; ret=val_in<120?60:val_in-60; return ret; </pre>

ID	Groovy Script Name	Groovy Script Text
286	Improve by 120	Double ret; ret=val_in<180?60:val_in-120; return ret;
285	Improve to - 60	Double ret = 60.0; return ret;
283	Improve to - 0	Double ret = 0; return ret;
276	Improve by 60 (min 80)	Double ret; ret=val_in<140?80:val_in-60; return ret;
271	Improve by 130	Double ret; ret=val_in<190?60:val_in-130; return ret;
270	Improve by 100	Double ret; ret=val_in<160?60:val_in-100; return ret;
267	Improve to - 0.13	Double ret; ret=val_in<0?0.13:val_in-0.13; return ret;
266	Improve to - 1.000	Double ret = 1.0000; return ret;
265	Improve to - 1.750	Double ret = 1.75; return ret;
264	Improve to - 100	Double ret = 100.0; return ret;
261	Improve by 45 (Min 100)	Double ret; ret=val_in<145?100:val_in-45; return ret;
260	Improve by 20	Double ret; ret=val_in<80?60:val_in-20; return ret;
-28	General: Improve by 10 Points	Double ret; ret=val_in>90?100.0:val_in+10.0; return ret;
-51	General: Set to 2	Double ret = 2; return ret;
154	No Improvement	Double ret; ret=val_in; return ret;

ID	Groovy Script Name	Groovy Script Text
111	Add 6 (0.3) to current value	Double ret; ret=val_in>4.7?5.0:val_in+0.3; return ret;
56	Improve to 90 (4.5)	Double ret; ret=4.5; return ret;
38	Improve to 100 (5)	Double ret = 5; return ret;
63	Add 20 (1) to current value	Double ret; ret=val_in>4?100:val_in+1; return ret;
62	Add 30 (1.5) to current value	Double ret; ret=val_in>3.5?100:val_in+1.5; return ret;
61	Add 5 (0.25) to current value	Double ret; ret=val_in>4.75?100:val_in+0.25; return ret;
60	Add 10 (0.5) to current value	Double ret; ret=val_in>4.5?5:val_in+0.5; return ret;
59	Improve to 70 (3.5)	Double ret; ret = 3.5; return ret;
84	import config log id	//args[0] is string to concatenate with if(args!=null && args.size()>0) { return ""+args[0]+import_config_log_id; } else { return import_config_log_id; }
332	PaveM mm to feet (Arg1 *	//args[0] is column Double millimeters = dr.getItemNumber(args[0]); if(millimeters==null) { return null; } else { return (millimeters*0.003280839); } }
333	Assign reversed direction	//this should use a case statement mapping to convert string to direction, then lookup by route if it should be reversed

ID	Groovy Script Name	Groovy Script Text
		<pre> //args[0] is the route column //args[1] is an expression or column //args[2] is an expression or column //args[3] is returned expression or column if args[1]==args[2] //args[4] is returned expression or column otherwise import com.agileassetsinc.core.*; if(MyClass.getScript() == null) { //check each arg if it is column or not List<String> argList = args as List; for(int i=1; i<argList.size(); i++) { if(argList.get(i)==null) { argList.set(i,""+null); continue; } if(dr.isValidColumn(argList.get(i))) { argList.set(i, "dr.getItemObject(\""+argList.get(i)+"\")"); } } String caseExpression = ""; for(int i=2; i<argList.size()-1; i=i+2) { caseExpression += "("+argList[1]+").equals(\""+argList[i]+") (\""+argList[1]+")==(\""+argList[i]+") ? (\""+argList[i+1]+") : "; } caseExpression += "("+argList[argList.size()-1]+")"; GroovyShell shell = new GroovyShell(); Script script = shell.parse(caseExpression); MyClass.setScript(script); } Script storedExpression = MyClass.getScript(); storedExpression.setBinding(this.getBin ding()); Integer dir = storedExpression.run(); return MyClass.processDir(dl, dr.getItemString(args[0]), dir); class MyClass { static DataStore dsReversedRoutes; static Script script; static getScript() { return script; } static setScript(Script script) { this.script = script; } static Integer processDir(DataLayer dl, String route, Integer dir) { if(this.dsReversedRoutes==null) { String routesSQL = "select substr </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (route_name,1,length(route_name)-1) as ROUTE from setup_network_lines where asc_dir in (2,4)"; this.dsReversedRoutes = dl.createDataStore(routesSQL); this.dsReversedRoutes.retrieve(); this.dsReversedRoutes.buildIndex("idx", "ROUTE"); } if(this.dsReversedRoutes.findByIndex("i dx", route) > 0) { switch(dir) { case 1: dir = 2; break; case 2: dir = 1; break; } } return dir; } </pre>
331	PaveM meters to feet (Arg1 *	<pre> //args[0] is column Double millimeters = dr.getItemNumber(args[0]); if(millimeters==null) { return null; } else { return (millimeters*3.2808399); } </pre>
330	PaveM mm to inches (Arg1 *	<pre> //args[0] is column Double millimeters = dr.getItemNumber(args[0]); if(millimeters==null) { return null; } else { return (millimeters*0.0393700787); } </pre>
138	FileToBlob_PDF	<pre> import com.agileassetsinc.core.*; File file = new File("/mnt/Import_mnt/1317/core/pdf/" + dr .getItemString(args[0]) + ".pdf"); // return file.exists(); TRDIBlob fileBlob = new TRDIBlob(TRDIUtils.fileToBytes(file)); String file_name = dr.getItemString(args[0]) + ".pdf"; String source_table = "PMS_CA_CORE"; String commentContentType = "application/pdf"; Integer comment_id = dl.getMaxId("COMMENTS", </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> DataStore dsComments = dl.createRetrieveDataStore("select * from COMMENTS where 1=2"); dsComments.setUpdateTable("COMMENTS"); Integer rownum_1 = dsComments.appendRow(); dsComments.setItemNumber(rownum_1 ,"COMMENT_ID",comment_id); dsComments.setItemString(rownum_1 ,"SOURCE_TABLE",source_table); dl.saveData(dsComments); DataStore ds = dl.createRetrieveDataStore("select * from COMMENTS_DET where 1=2"); ds.setUpdateTable("COMMENTS_DET"); Integer rownum = ds.appendRow(); ds.setItemNumber(rownum ,"COMMENT_ID",comment_id); ds.setItemString(rownum,"COMMENT_PATH",file_name); ds.setItemString(rownum,"COMMENT_CONTENT_TYPE",commentContentType); ds.setItemBlob(rownum ,"COMMENT_DATA",fileBlob); ds.setItemString(rownum ,"SOURCE_TABLE",source_table); dl.saveData(ds); return comment_id; </pre>
321	Concatenate 2 Strings w/(-)	<pre> //args[0] is column return (dr.getItemObject(args[0])+"-"+dr.getItemObject(args[1])); </pre>
254	Set Max loc_ident if source column	<pre> //args[0] is step import com.agileassetsinc.core.*; if(args!=null) return null; Integer step=1000; return LocIdentSequence.getNext(dl, step); class LocIdentSequence { static Long loc_ident; static Long max_loc_ident; static getNext(DataLayer dl, int step) { if(loc_ident==null loc_ident>=max_loc_ident) { loc_ident = dl.getMaxId("SETUP_LOC_IDENT", "LOC_IDENT", step); max_loc_ident = </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> loc_ident+step-1; } else { loc_ident++; } } return loc_ident; } } </pre>
253	Parse CA Bridge Location for	<pre> //arg should be Route Column //should be something like 01-HUM-254-17.8-STPK //PM prefix or suffix may not be there also last - may not be there import com.agileassetsinc.core.*; String postmile; String loc= dr.getItemString(args[0]); if(loc==null loc.length()==0) return null; if(TRDIUtils.instr(loc,'-',0,4)==-1) { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1); } else { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1, TRDIUtils.instr(loc,'-',0,4)); } String prefix = getPrefix(postmile); if(prefix!=null) postmile = postmile.substring(prefix.length()); String suffix = getSuffix(postmile); if(suffix !=null) postmile = postmile.substring(0, postmile.length()- suffix .length()); return postmile; String getSuffix(String loc) { if(loc==null loc.length()<=1) return null; String suffix= loc.substring(loc.length()- 2); try { new Double(suffix); return null; } catch(Exception e){ return suffix; }; } String getPrefix(String loc) { if(loc==null loc.length()==0) return null; String prefix = loc.substring(0,1); if(prefix==".") prefix=0; try { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> new Integer(prefix); return null; } catch(Exception e){ return prefix, }; } </pre>
252	Parse CA Rte Suffix from Bridge	<pre> //arg should be Route Column //should be something like 01-HUM-254-17.8-STPK //PM prefix or suffix may not be there also last - may not be there import com.agileassetsinc.core.*; String loc= dr.getItemString(args[0]); if(loc==null loc.length()==0) return null; String route = loc.substring(TRDIUtils.instr(loc,'-',0,2)+1, TRDIUtils.instr(loc,'-',0,3)); String rteSuffix= getRteSuffix(route); return rteSuffix; String getRteSuffix(String loc) { if(loc==null loc.length()==0) return null; String suffix= loc.substring(loc.length()-2); try { new Integer(suffix); return null; } catch(Exception e){ return suffix; } } </pre>
251	Parse CA Route from bridge	<pre> //arg should be Route Column //should be something like 01-HUM-254-17.8-STPK //PM prefix or suffix may not be there also last - may not be there import com.agileassetsinc.core.*; String loc= dr.getItemString(args[0]); if(loc==null loc.length()==0) return null; String route = loc.substring(TRDIUtils.instr(loc,"-",0,2)+1, TRDIUtils.instr(loc,"-",0,3)); String rteSuffix= getRteSuffix(route); if(rteSuffix !=null) route = route. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> substring(0, route.length()- rteSuffix.length()); return route; String getRteSuffix(String loc) { if(loc==null loc.length()==0) return null; String suffix= loc.substring(loc.length()- 2); try { new Integer(suffix); return null; } catch(Exception e){ return suffix; } }; } </pre>
250	Parse CA Bridge Location PM Suffix	<pre> //arg should be Route Column //should be something like 01-HUM-254- 17.8-STPK //PM prefix or suffix may not be there also last - may not be there import com.agileassetsinc.core.*; String postmile; String loc= dr.getItemString(args[0]); if(loc==null loc.length()==0) return null; if(TRDIUtils.instr(loc,'-',0,4)==-1) { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1); } else { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1, TRDIUtils.instr(loc,'-',0,4)); } String prefix = getPrefix(postmile); if(prefix!=null) postmile = postmile.substring(prefix.length()); String suffix = getSuffix(postmile); if(suffix !=null) postmile = postmile.substring(0, postmile.length()- suffix .length()); return suffix; String getSuffix(String loc) { if(loc==null loc.length()<=1) return null; String suffix= loc.substring(loc.length()- 2); try { new Double(suffix); return null; } catch(Exception e){ </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> return suffix; }; } String getPrefix(String loc) { if(loc==null loc.length()==0) return null; String prefix = loc.substring(0,1); if(prefix==".") prefix=0; try { new Integer(prefix); return null; } catch(Exception e){ return prefix; } }; } </pre>
248	Parse CA Bridge Location PM Prefix	<pre> //arg should be Route Column //should be something like 01-LAK-053-1 //PM prefix or suffix may not be there also last - may not be there import com.agileassetsinc.core.*; String postmile; String loc= dr.getItemString(args[0]); if(loc==null loc.length()==0) return null; if(TRDIUtils.instr(loc,'-',0,4)==-1) { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1); } else { postmile = loc.substring(TRDIUtils.instr(loc,'-',0,3)+1, TRDIUtils.instr(loc,'-',0,4)); } String prefix = getPrefix(postmile); if(prefix!=null) postmile = postmile.substring(prefix.length()); String suffix = getSuffix(postmile); if(suffix !=null) postmile = postmile.substring(0, postmile.length()- suffix .length()); return prefix; String getSuffix(String loc) { if(loc==null loc.length()<=1) return null; String suffix= loc.substring(loc.length()- 2); try { new Double(suffix); return null; } catch(Exception e){ return suffix; } }; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } String getPrefix(String loc) { if(loc==null loc.length()==0) return null; String prefix = loc.substring(0,1); if(prefix==".") prefix=0; try { new Double(prefix); return null; } catch(Exception e){ return prefix, }; } </pre>
246	Parse CA Bridge Location County	<pre> //arg should be Bridge location //should be something like 01-HUM-254- 17.8-STPK //County HUM //PM prefix or suffix may not be there import com.agileassetsinc.core.*; String loc= dr.getItemString(args[0]); String county=getCounty(loc); if(county==null county.length()==0) return null; String sql_str="select A.CA_COUNTY_ID from setup_ca_county a where A.CA_COUNTY_ABBREV='"+county+"' "; DataStore ds = dl.createRetrieveDataStore(sql_str); return ds.getItemInteger(1,"CA_COUNTY_ID"); String getCounty(String loc) { if(loc==null loc.length()==0) return null; String county= (loc.substring(5,6)=="-") ? loc.substring(3,5) : loc.substring(3,6); return county ; } </pre>
245	Concatenate 3 strings	<pre> //args[0] is column return ((dr.isItemNull(args[0]) ? "" : dr.getItemObject(args[0]))+" "+(dr.isItem Null(args[1]) ? "" : dr.getItemObject(args[1]))+" "+(dr.isItem Null(args[2]) ? "" : dr.getItemObject(args[2]))); </pre>
244	Parse Ca Route and Rte Suffix	<pre> //arg should be Route Column //should be something like 501, or 101T, or 201S </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> //suffix may or may not be there import com.agileassetsinc.core.*; String route= dr.getItemString(args[0]); String suffix = getSuffix(route); if(suffix !=null) route= route.substring(0, route.length()-suffix .length()); if(target_column_id.equals("CA_RTE_S UFFIX_FROM") target_column_id.equals("CA_RTE_SUF FIX_TO")) { return suffix ; } else if(target_column_id.equals("CA_ROUTE _FROM") target_column_id.equals("CA_ROUTE_ TO")) { if(route==null route.length()==0) return null; else return new Double(route); } String getSuffix(String route) { if(route==null route.length()==0) return null; String suffix = route.substring(route.length()-1); try { new Integer(suffix); return null; } catch(Exception e){ return suffix ; } }; } </pre>
243	parse from Post Miles	<pre> //arg should be Post Miles column //should be something like M21.847, or 6.601, or R30.584R //neither prefix or suffix are always there. import com.agileassetsinc.core.*; String postmile = dr.getItemString(args[0]); String prefix = getPrefix(postmile); if(prefix!=null) postmile = postmile.substring(prefix.length()); String suffix = getSuffix(postmile); if(suffix !=null) postmile = postmile.substring(0, postmile.length()- suffix .length()); if(target_column_id.equals("CA_PM_PR EFIX_FROM") target_column_id.equals("CA_PM_PRE FIX_TO")) { return prefix; } else if(target_column_id.equals("CA_PM_SU FFIX_FROM") </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> target_column_id.equals("CA_PM_SUF FIX_TO")) { return suffix ; } else if(target_column_id.equals("CA_POSTM ILE_BEG") target_column_id.equals("CA_POSTMIL E_END")) { if(postmile==null postmile.length()==0) return null; else return new Double(postmile); } String getPrefix(String postmile) { if(postmile==null postmile.length()==0) return null; String prefix = postmile.substring(0,1); try { new Integer(prefix); return null; } catch(Exception e){ return prefix; } }; String getSuffix(String postmile) { if(postmile==null postmile.length()==0) return null; String suffix = postmile.substring(postmile.length()-1); try { new Integer(suffix); return null; } catch(Exception e){ return suffix ; } }; </pre>
-1062	General: sync by column name	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return dl.getMaxId(ds.getBaseTable(), target_column_id); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+"; "+ds_col.getItemString(1,"DESCR_COL _NAME")+"; from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.setMaxRows(0); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds; } } </pre>
-1061	General: import config log id	<pre> //args[0] is string to concatenate with if(args!=null && args.size()>0) { return ""+args[0]+import_config_log_id; } else { return import_config_log_id; } </pre>
-1060	General: yes/no/null	<pre> //args[0] is column import com.agileassetsinc.core.*; if(args==null args.length<1) { throw new TRDException("No Argument Passed to Script"); } String val = dr.getItemString(args[0]); if(val==null) { return null; } else if(val.equalsIgnoreCase("TRUE") val.equalsIgnoreCase("T") val.equalsIgnoreCase("YES") val.equalsIgnoreCase("Y") val.equalsIgnoreCase("1")) { return 1; } else if(val.equalsIgnoreCase("FALSE") val.equalsIgnoreCase("F") val.equalsIgnoreCase("NO") val.equalsIgnoreCase("N") val. </pre>

ID	Groovy Script Name	Groovy Script Text
-57	General: yes/no	<pre> equalsIgnoreCase("0")) { return 0; } else { throw new TRDException("'" + val + "' Is Invalid Value"); } //args[0] is column import com.agileassetsinc.core.*; if(args==null args.length<1) { throw new TRDException("No Argument Passed to Script"); } String val = dr.getItemString(args[0]); if(val==null) { throw new TRDException("'null' Is Invalid Value"); } else if(val.equalsIgnoreCase("TRUE") val.equalsIgnoreCase("T") val.equalsIgnoreCase("YES") val.equalsIgnoreCase("Y") val.equalsIgnoreCase("1")) { return 1; } else if(val.equalsIgnoreCase("FALSE") val.equalsIgnoreCase("F") val.equalsIgnoreCase("NO") val.equalsIgnoreCase("N") val.equalsIgnoreCase("0")) { return 0; } else { throw new TRDException("'" + val + "' Is Invalid Value"); } </pre>
-56	General: year from date	<pre> //args[0] is column //args[1] is format //default format is "yyyy/MM/dd HH:mm:ss" /* Letter Date or Time Component Presentation Examples G Era designator Text AD y Year Year 1996; 96 M Month in year Month July; Jul; 07 w Week in year Number 27 W Week in month Number 2 D Day in year Number 189 d Day in month Number 10 F Day of week in month Number 2 E Day in week Text Tuesday; Tue a Am/pm marker Text PM H Hour in day (0-23) Number 0 k Hour in day (1-24) Number 24 K Hour in am/pm (0-11) Number 0 h Hour in am/pm (1-12) Number 12 m Minute in hour Number 30 </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> s Second in minute Number 55 S Millisecond Number 978 z Time zone General time zone Pacific Standard Time; PST; GMT-08:00 Z Time zone RFC 822 time zone - 0800 */ import com.agileassetsinc.core.*; if(args!=null && args.length>0) { if(dr.getItemObject(args[0]) instanceof DateTime) { return dr.getItemObject(args[0]).getYear(); } else { String format = args.length>1 ? args[1] : null; date = TRDIUtils.strToDate(dr.getItemObject(ar gs[0]),format); return (date!=null ? date.getYear() : null); } } else { return new DateTime().getYear(); } } //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return dl.getMaxId(ds.getBaseTable(), target_column_id); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDIException(target_column_id+" is not T-Type </pre>
235	sync by column name	

ID	Groovy Script Name	Groovy Script Text
		<pre> column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.setMaxRows(0); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));/idx on name ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds; } </pre>
-33	General: map by column name insert	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { int row = ds.appendRow(); ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, dr.getItemString(args[0])); dl.saveData(ds); return ds.getItemObject(row,1); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDIEException(target_column_id+" is not T-Type column"); } } } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME")); //idx on name ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds; } </pre>
-32	General: map by column name (H-	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1) { throw new TRDIEException(target_column_id+" is not present in setup_column_id"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+ (ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE")); ds.retrieve(); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } } </pre>
-30	General: map by column comments	<pre> import com.agileassetsinc.core.*; //args[0] is column ds = MyDs.getDs(dl, target_column_id); String val = dr.getItemString(args[0]); if(val!=null) val = val.trim(); int[] rows = ds.findAllByIndex("idx",val); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", COMMENT_STR from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx","COMMENT_STR");// idx on comment_str } return ds; } } </pre>
-27	General: evaluate expression	<pre> //args[] is an expression or column import com.agileassetsinc.core.*; //check each arg if it is column or not argList = args as List; for(int i=0; i<argList.size(); i++) { </pre>

ID	Groovy Script Name	Groovy Script Text
-26	General: Counter (restart at initial	<pre> if(dr.isValidColumn(argList.get(i))) { value = dr.getItemObject(argList.get(i)); argList.set(i, (value==null ? "" : value instanceof String ? "\"" + value + "\"" : value)); } } return evaluate(argList.join("")); //args[0] is start_value import com.agileassetsinc.core.*; if(args==null) start_value = 1; else start_value = new Integer(args[0]); return MyCounter.getNext(start_value); class MyCounter { static Long value; static getNext(int start_value) { if(value==null) { value = start_value; } else { value++; } return value; } } </pre>
-25	General: Concatenate 5 Strings	<pre> //args[0] is column return (dr.getItemObject(args[0])+""+dr.getItem Object(args[1])+""+dr.getItemObject(arg s[2])+""+dr.getItemObject(args[3])+""+dr. getItemObject(args[4])); </pre>
-22	General: case statement	<pre> //args[0] is an expression or column //args[1] is an expression or column //args[2] is returned expression or column if args[0]==args[1] //args[3] is returned expression or column otherwise import com.agileassetsinc.core.*; if(MyExpression.getScript() == null) { //check each arg if it is column or not argList = args as List; for(int i=0; i<argList.size(); i++) { if(argList.get(i)==null) { argList.set(i, ""+null); continue; } if(dr.isValidColumn(argList.get(i))) { argList.set(i, "dr.getItemObject(\""+argList.get(i)+"\""); } } } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> String caseExpression = ""; for(int i=1; i<argList.size()-1; i=i+2) { caseExpression += "("+argList[0]+").equals("("+argList[i]+")) "("+argList[0]+")==("("+argList[i]+") ? "("+argList[i+1]+") : "; } caseExpression += "("+argList[argList.size()-1]+")"; GroovyShell shell = new GroovyShell(); script = shell.parse(caseExpression); MyExpression.setScript(script); } Script storedExpression = MyExpression.getScript(); storedExpression.setBinding(this.getBinding()); return storedExpression.run(); class MyExpression { static Script script; static getScript() { return script; } static setScript(Script script) { this.script = script; } } </pre>
-54	General: today	<pre> //no arguments import com.agileassetsinc.core.*; return new DateTime(); </pre>
-52	General: set to column value	<pre> //args[0] is column return dr.getItemObject(args[0]); </pre>
-41	General: next in sequence	<pre> //args[0] is step import com.agileassetsinc.core.*; if(args==null) step = 1000; else step = new Integer(args[0]); return MySequence.getNext(dl, target_column_id, step); class MySequence { static Long value; static Long max_value; static String table_name; static String column_name; static getNext(DataLayer dl, String target_column_id, int step) { if(value==null value>=max_value) { if(table_name==null column_name==null) { DataStore ds_col = dl.createDataStore("select * from </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> setup_column_id where column_id = ?"); if(ds_col.retrieve(target_column_id)==1) { table_name = ds_col.getItemString(1,"TABLE_NAME") ; column_name = ds_col.getItemString(1,"ITEM_COL_NAME"); } if(table_name==null) { throw new TRDIEException("Invalid table sequence"); } else if(table_name.toUpperCase().endsWith(" INVENTORY")) { table_name = "IDENT_INV_ELEMENT"; column_name = "INV_ELEM_ID"; } } value = dl.getMaxId(table_name, column_name, step); max_value = value+step-1; } else { value++; } } return value; } </pre>
-31	General: map by column name	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (target_column_id); if(ds_col.rowCount())<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+"," "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+(ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE"))); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } </pre>
-38	General: max loc_ident	<pre> //args[0] is step import com.agileassetsinc.core.*; if(args==null) step = 1000; else step = new Integer(args[0]); return LocIdentSequence.getNext(dl, step); class LocIdentSequence { static Long loc_ident; static Long max_loc_ident; static getNext(DataLayer dl, int step) { if(loc_ident==null loc_ident>=max_loc_ident) { loc_ident = dl.getMaxId("SETUP_LOC_IDENT", "LOC_IDENT", step); max_loc_ident = loc_ident+step-1; } else { loc_ident++; } return loc_ident; } } </pre>
-37	General: map by multiple	<pre> //args[0] is lookup table //args[1] is the column needed to return //args[2],arg[3]... is source,lookup column(s) to match to find the return </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, args); String[] idxCols = MyDs.getIdxCols(); Object[] idxArgs = new Object[idxCols.length]; for(int i=0; i<idxCols.length; i++) { idxArgs[i] = dr.getItemObject(idxCols[i]); } int[] rows = ds.findAllByIndex("idx",idxArgs); if(rows.length>1) throw new TRDException(target_column_id+", Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { throw new TRDException(target_column_id+", Target not found"); } class MyDs { static DataStore ds; static String[] sourceCols;//columns in source used to find by index static String[] lookupCols;//columns in lookup used to find by index static getDs(DataLayer dl, String[] args) { if(ds==null) { StringBuffer sb = new StringBuffer(); sourceCols = new String[(args.length-2)/2]; lookupCols = new String[(args.length-2)/2]; for(int i=1; i<=(args.length-2)/2; i++) { sb.append(sb.length()>0 ? "," : "").append(args[2*i+1]); sourceCols[i-1] = args[2*i]; lookupCols[i-1] = args[2*i+1]; } ds = dl.createDataStore("select "+args[1]+",""+sb.toString()+" from "+args[0]); ds.retrieve(); ds.buildIndex("idx",lookupCols);//idx on lookupCols } return ds; } static getIdxCols() { </pre>

ID	Groovy Script Name	Groovy Script Text
-53	General: to date	<pre> return sourceCols, } } //args[0] is column //args[1] is format //default format is "yyyy/MM/dd HH:mm:ss" /* Letter Date or Time Component Presentation Examples G Era designator Text AD y Year Year 1996; 96 M Month in year Month July; Jul; 07 w Week in year Number 27 W Week in month Number 2 D Day in year Number 189 d Day in month Number 10 F Day of week in month Number 2 E Day in week Text Tuesday; Tue a Am/pm marker Text PM H Hour in day (0-23) Number 0 k Hour in day (1-24) Number 24 K Hour in am/pm (0-11) Number 0 h Hour in am/pm (1-12) Number 12 m Minute in hour Number 30 s Second in minute Number 55 S Millisecond Number 978 z Time zone General time zone Pacific Standard Time; PST; GMT-08:00 Z Time zone RFC 822 time zone - 0800 */ import com.agileassetsinc.core.*; if(args!=null && args.length>0) { String format = args.length>1 ? args[1] : null; return TRDIUtils.strToDate(dr.getItemObject(ar gs[0]),format); } else { return new DateTime(); } </pre>
-36	General: map inv_elem_id by name	<pre> //args[0] is column //args[1] is asset type import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id, args[1]); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIException("Multiple found"); if(rows[0]) { return ds.getItemObject(rows[0],"ID"); } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } else { throw new TRDException("Target not found"); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id, String asset_type_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+ " as id, "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " as name from "+ds_col.getItemString(1,"TABLE_NAM E")+ " where asset_type_id="+asset_type_id); ds.setMaxRows(0); ds.retrieve(); ds.buildIndex("idx","NAME");//idx on name } return ds; } } </pre>
-35	General: map by column name	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { throw new TRDException("Target not found"); } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDIEException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+(ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE")); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } } </pre>
-34	General: map by column name insert	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemString(args[0]).trim()); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { if(ds.rowCount()<1) {//add root if none exists int row = ds.appendRow(); ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, "New"); dl.saveData(ds); } int row = ds.appendRow(); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, dr.getItemString(args[0]).trim()); ds.setItemNumber(row, 3, ds.getItemInteger(1,1)); dl.saveData(ds); return ds.getItemObject(row,1); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+", "+ds_col.getItemString(1,"PARENT_CO L_NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME")); //idx on name ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds, } } </pre>
-19	General: Addition (Arg1 + Arg2)	<pre> //args[0] is column return (dr.getItemNumber(args[0])+dr.getItemN </pre>
141	map by column name required _	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { throw new TRDIEException("Target not found"); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDIEException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+(ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE"))); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } } </pre>
140	nextEventID	<pre> import com.agileassetsinc.core.*; Integer unique_id = dl.getMaxId("ID_SIGN_EVENT_HIST", "ID_SIGN_EVENT_HIST_ID"); </pre>
31	year from date	<pre> return unique_id; //args[0] is column //args[1] is format //default format is "yyyy/MM/dd HH:mm:ss" /* Letter Date or Time Component </pre>

ID	Groovy Script Name	Groovy Script Text
		<p>Presentation Examples</p> <p>G Era designator Text AD</p> <p>y Year Year 1996; 96</p> <p>M Month in year Month July; Jul; 07</p> <p>w Week in year Number 27</p> <p>W Week in month Number 2</p> <p>D Day in year Number 189</p> <p>d Day in month Number 10</p> <p>F Day of week in month Number 2</p> <p>E Day in week Text Tuesday; Tue</p> <p>a Am/pm marker Text PM</p> <p>H Hour in day (0-23) Number 0</p> <p>k Hour in day (1-24) Number 24</p> <p>K Hour in am/pm (0-11) Number 0</p> <p>h Hour in am/pm (1-12) Number 12</p> <p>m Minute in hour Number 30</p> <p>s Second in minute Number 55</p> <p>S Millisecond Number 978</p> <p>z Time zone General time zone Pacific Standard Time; PST; GMT-08:00</p> <p>Z Time zone RFC 822 time zone - 0800</p> <p>*/</p> <pre> import com.agileassetsinc.core.*; if(args!=null && args.length>0) { if(dr.getItemObject(args[0]) instanceof DateTime) { return dr.getItemObject(args[0]).getYear(); } else { String format = args.length>1 ? args[1] : null; date = TRDIUtils.strToDate(dr.getItemObject(ar gs[0]),format); return (date!=null ? date.getYear() : null); } } else { return new DateTime().getYear(); } </pre>
30	case statement	<pre> //args[0] is an expression or column //args[1] is an expression or column //args[2] is returned expression or column if args[0]==args[1] //args[3] is returned expression or column otherwise import com.agileassetsinc.core.*; //check each arg if it is column or not argList = args as List; for(int i=0; i<argList.size(); i++) { if(argList.get(i)==null) { argList.set(i,""+null); continue; } } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dr.isValidColumn(argList.get(i))) { value = dr.getItemObject(argList.get(i)); //string_or_null = dr.getColumn(argList.get(i)).isString(); if(dr.getColumn(argList.get(i)).isString() && value!=null) { argList.set(i, "\"" +value+"\""); } else { argList.set(i, ""+value); } } return caseStatement(argList); def caseStatement(List args) { if(args.size==2) { return evaluate(args[1]); } else if(evaluate(args[0]).equals(evaluate(args [1])) evaluate(args[0])==evaluate(args[1])) { return evaluate(args[2]); } else { args.remove(2); args.remove(1); return caseStatement(args); } //args[0] is column import com.agileassetsinc.core.*; if(args==null args.length<1) { throw new TRDException("No Argument Passed to Script") } String val = dr.getItemString(args[0]); if(val == null) { return 0; } if(val.equalsIgnoreCase("TRUE") val.equalsIgnoreCase("T") val.equalsIgnoreCase("YES") val.equalsIgnoreCase("Y") val.equalsIgnoreCase("1")) { return 1; } else if(val.equalsIgnoreCase("FALSE") val.equalsIgnoreCase("F") val.equalsIgnoreCase("NO") val.equalsIgnoreCase("N") val.equalsIgnoreCase("0")) { return 0; } else { return null; } </pre>
29	yes/no	

ID	Groovy Script Name	Groovy Script Text
28	today	//no arguments
27	to date	<pre> import com.agileassetsinc.core.*; return new DateTime(); //args[0] is column //args[1] is format //default format is "yyyy/MM/dd HH:mm:ss" /* Letter Date or Time Component Presentation Examples G Era designator Text AD y Year Year 1996; 96 M Month in year Month July; Jul; 07 w Week in year Number 27 W Week in month Number 2 D Day in year Number 189 d Day in month Number 10 F Day of week in month Number 2 E Day in week Text Tuesday; Tue a Am/pm marker Text PM H Hour in day (0-23) Number 0 k Hour in day (1-24) Number 24 K Hour in am/pm (0-11) Number 0 h Hour in am/pm (1-12) Number 12 m Minute in hour Number 30 s Second in minute Number 55 S Millisecond Number 978 z Time zone General time zone Pacific Standard Time; PST; GMT-08:00 Z Time zone RFC 822 time zone - 0800 */ import com.agileassetsinc.core.*; if(args!=null && args.length>0) { String format = args.length>1 ? args[1] : null; return TRDIUtils.strToDate(dr.getItemObject(ar gs[0]),format); } else { return new DateTime(); } </pre>
136	convert_fileToBlob	<pre> import com.agileassetsinc.core.*; File file = new File("/mnt/Castor_mnt/1317/HIMS_ITD/S IGN_IMG_JPG/"+dr.getItemString(args[0])+".jpg"); // return file.exists(); TRDIBlob fileBlob = new TRDIBlob(TRDIUtils.fileToBytes(file)); // im not sure what you are trying to do here... but you should be able to just return the blob and it should </pre>

ID	Groovy Script Name	Groovy Script Text
		be set for the column... DataStore ds = dl.createDataStore(fileBlob); ds.retrieve(); dsIn.setItemBlob(target_column_id,fileBlob);
137	fileToBlob	// return fileBlob; import com.agileassetsinc.core.*; File file = new File("/mnt/Import_mnt/1317/core/images/ D1/"+dr.getItemString(args[0])+".jpg"); // return file.exists(); TRDIBlob fileBlob = new TRDIBlob(TRDIUtils.fileToBytes(file)); String file_name = dr.getItemString(args[0]) + ".jpg"; String source_table = "PMS_CA_CORE"; String commentContentType = "image/jpeg"; Integer comment_id = dl.getMaxId("COMMENTS", "COMMENT_ID"); DataStore dsComments = dl.createRetrieveDataStore("select * from COMMENTS where 1=2"); dsComments.setUpdateTable("COMMENTS"); Integer rownum_1 = dsComments.appendRow(); dsComments.setItemNumber(rownum_1 ,"COMMENT_ID",comment_id); dsComments.setItemString(rownum_1 ,"SOURCE_TABLE",source_table); dl.saveData(dsComments); DataStore ds = dl.createRetrieveDataStore("select * from COMMENTS_DET where 1=2"); ds.setUpdateTable("COMMENTS_DET"); Integer rownum = ds.appendRow(); ds.setItemNumber(rownum ,"COMMENT_ID",comment_id); ds.setItemString(rownum,"COMMENT_PATH",file_name); ds.setItemString(rownum,"COMMENT_CONTENT_TYPE",commentContentType); ds.setItemBlob(rownum ,"COMMENT_DATA",fileBlob); ds.setItemString(rownum ,"SOURCE_TABLE",source_table); dl.saveData(ds);

ID	Groovy Script Name	Groovy Script Text
34	map by column comment	<pre> return comment_id; import com.agileassetsinc.core.*; //args[0] is column ds = MyDs.getDs(dl, target_column_id); String val = dr.getItemString(args[0]); if(val!=null) val = val.trim(); int[] rows = ds.findAllByIndex("idx",val); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", COMMENT_STR from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx","COMMENT_STR");// idx on comment_str } return ds; } } </pre>
33	map by trimmed column name	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0]).trim()); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+(ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE"))); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } } </pre>
32	evaluate expression	<pre> //args[] is an expression or column import com.agileassetsinc.core.*; //check each arg if it is column or not argList = args as List; for(int i=0; i<argList.size(); i++) { if(dr.isValidColumn(argList.get(i))) { value = dr.getItemObject(argList.get(i)); argList.set(i, (value==null ? "" : value instanceof String ? "\"" +value+"\"" : value)); } } return evaluate(argList.join("")); </pre>

ID	Groovy Script Name	Groovy Script Text
26	map by column name insert	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { int row = ds.appendRow(); ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, dr.getItemString(args[0])); dl.saveData(ds); return ds.getItemObject(row,1); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1!ds_col.getItemSt ring(1,"VIEW_TYPE").equals("T")) { throw new TRDIEException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+ "+ds_col.getItemString(1,"DESCR_COL _NAME")+ "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME")); ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds; } } </pre>

ID	Groovy Script Name	Groovy Script Text
25	map by column name	<pre>//args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { return null; } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+(ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1,"WHERE_CLA USE"))); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));/idx on name } return ds; } }</pre>
6	set to column value	<pre>//args[0] is column return dr.getItemObject(args[0]);</pre>
5	next in sequence	<pre>//args[0] is step import com.agileassetsinc.core.*;</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(args==null) step = 1000; else step = new Integer(args[0]); return MySequence.getNext(dl, target_column_id, step); class MySequence { static Long value; static Long max_value; static String table_name; static String column_name; static getNext(DataLayer dl, String target_column_id, int step) { if(value==null value>=max_value) { if(table_name==null column_name==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); if(ds_col.retrieve(target_column_id)==1) { table_name = ds_col.getItemString(1,"TABLE_NAME") ; column_name = ds_col.getItemString(1,"ITEM_COL_NA ME"); } if(table_name==null) { throw new TRDIException("Invalid table sequence"); } else if(table_name.toUpperCase().endsWith(" INVENTORY")) { table_name = "IDENT_INV_ELEMENT"; column_name = "INV_ELEM_ID"; } } value = dl.getMaxId(table_name, column_name, step); max_value = value+step-1; } else { value++; } return value; } } </pre>
4	max loc_ident	<pre> //args[0] is step import com.agileassetsinc.core.*; if(args==null) step = 1000; else step = new Integer(args[0]); return LocIdentSequence.getNext(dl, step); class LocIdentSequence { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> static Long loc_ident; static Long max_loc_ident; static getNext(DataLayer dl, int step) { if(loc_ident==null loc_ident>=max_loc_ident) { loc_ident = dl.getMaxId("SETUP_LOC_IDENT", "LOC_IDENT", step); max_loc_ident = loc_ident+step-1; } else { loc_ident++; } return loc_ident; } </pre>
3	map inv_elem_id by name required	<pre> //args[0] is column //args[1] is asset type import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id, args[1]); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDIEException("Multiple found"); if(rows[0]) { return ds.getItemObject(rows[0],"ID"); } else { throw new TRDIEException("Target not found"); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id, String asset_type_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDIEException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> "+ds_col.getItemString(1,"ITEM_COL_NAME")+ " as id, "+ds_col.getItemString(1,"DESCR_COL_NAME")+ " as name from "+ds_col.getItemString(1,"TABLE_NAME")+ " where asset_type_id="+asset_type_id; ds.setMaxRows(0); ds.retrieve(); ds.buildIndex("idx","NAME");//idx on name } return ds, } } </pre>
2	map by column name required	<pre> //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemObject(args[0])); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { throw new TRDException("Target not found"); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount()<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException(target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+ ", "+ds_col.getItemString(1,"DESCR_COL _NAME")+ " from "+ds_col.getItemString(1,"TABLE_NAM E")+ (ds_col.isItemNull(1,"WHERE_CLA USE") ? "" : " where "+ds_col.getItemString(1," </pre>

ID	Groovy Script Name	Groovy Script Text
		WHERE_CLAUSE"))); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name } return ds; } } //args[0] is column import com.agileassetsinc.core.*; ds = MyDs.getDs(dl, target_column_id); int[] rows = ds.findAllByIndex("idx",dr.getItemString(args[0]).trim()); if(rows.length>1) throw new TRDException("Multiple found"); if(rows.length==1) { return ds.getItemObject(rows[0],1); } else { if(ds.rowCount(<1)){//add root if none exists int row = ds.appendRow(); ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, "New"); dl.saveData(ds); } int row = ds.appendRow(); ds.setItemNumber(row, 1, dl.getMaxId(ds.getBaseTable(), target_column_id)); ds.setItemString(row, 2, dr.getItemString(args[0]).trim()); ds.setItemNumber(row, 3, ds.getItemInteger(1,1)); dl.saveData(ds); return ds.getItemObject(row,1); } class MyDs { static DataStore ds; static getDs(DataLayer dl, String target_column_id) { if(ds==null) { DataStore ds_col = dl.createDataStore("select * from setup_column_id where column_id = ?"); ds_col.retrieve(target_column_id); if(ds_col.rowCount(<1 !ds_col.getItemString(1,"VIEW_TYPE"). equals("T")) { throw new TRDException
1	map by column name insert for tree	

ID	Groovy Script Name	Groovy Script Text
		<pre> (target_column_id+" is not T-Type column"); } ds = dl.createDataStore("select "+ds_col.getItemString(1,"ITEM_COL_N AME")+"; "+ds_col.getItemString(1,"DESCR_COL _NAME")+"; "+ds_col.getItemString(1,"PARENT_CO L_NAME")+"; from "+ds_col.getItemString(1,"TABLE_NAM E")); ds.retrieve(); ds.buildIndex("idx",ds_col.getItemString(1,"DESCR_COL_NAME"));//idx on name ds.setUpdateTable(ds_col.getItemString (1,"TABLE_NAME")); } return ds; } </pre>
53	MMS Model ams_id	<pre> //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script Double futureBudget=currentBudget; if (currentBudget!=null && currentBudget>0) { if(currentScore>0) { if(currentScore>99) currentScore = 99; if(targetScore>99) targetScore = 99; futureBudget = currentBudget * ((100-currentScore)/currentScore) * (targetScore/(100-targetScore)); } } return futureBudget; </pre>
-29	General: LOS_DEFECT_COEF	<pre> import com.agileassetsinc.core.*; Integer a1 = Math.floor(currentScore); Double s1 = 1.0; Double s2 = 1.0; Integer ownerNet = dsIn.getItemInteger(rowIn,"OWNER_NE T"); Integer functClassId = dsIn.getItemInteger(rowIn,"FUNCT_CLA SS_ID"); Double coef1=1.0; Double coef2=1.0; DataStore ds_coef1 = dl.createDataStore("select DEFECT_COEFF\n" + "from los_coeff_vw3\n" + </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> "where EFF_FISCAL_YEAR=?\n" + "and DEFECT_ID=?\n" + "and OWNER_NET=?\n" + "and FUNCT_CLASS_ID=?\n" + "and DEFECT_SCORE=?"); if(ds_coef1.retrieve(effFiscalYear,defectId,ownerNet,functClassId,a1)==1) { s1 = ds_coef1.getItemNumber(1,1); } if(ds_coef1.retrieve(effFiscalYear,defectId,ownerNet,functClassId,a1+1)==1) { s2 = ds_coef1.getItemNumber(1,1); } coef1=s1 + (currentScore - a1)*(s2 - s1); DataStore ds_coef2 = dl.createDataStore("select DEFECT_COEFF\n" + "from los_coef_vw_all_scores\n" + "where EFF_FISCAL_YEAR=?\n" + "and DEFECT_ID=?\n" + "and OWNER_NET=?\n" + "and FUNCT_CLASS_ID=?\n" + "and FROM_SCORE=?\n" + "and TO_SCORE=?"); if(ds_coef2.retrieve(effFiscalYear,defectId,ownerNet,functClassId,a1,targetScore) ==1) { coef2 = ds_coef2.getItemNumber(1,1); } return currentBudget/coef1*coef2; </pre>
-40	General: Exponential	<pre> import com.agileassetsinc.core.*; //////////////////////////////////// //Configuration Section //set these two variables Integer BestValue = 100; Integer WorstValue = 1; //////////////////////////////////// if(BestValue.equals(WorstValue)) { throw new TRDIException("Incorrect groovy script configuration, BestValue cannot equal WorstValue"); } Double futureBudget=currentBudget; if (currentBudget!=null && </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> currentBudget>0) { if(currentScore < 1) currentScore = 1; if(currentScore>99) currentScore = 99; if(targetScore==100) futureBudget = 2*currentBudget * ((100- currentScore)/currentScore) * (99); else futureBudget = currentBudget * ((100- currentScore)/currentScore) * (targetScore/(100-targetScore)); } </pre>
		<pre> return futureBudget; </pre>
162	Send e-mail (vm)	<pre> import javax.mail.internet.*; import javax.mail.* import javax.activation.* message = "test groovy mail" subject = "groovy sent this" toAddress = "Vince.Main@itd.idaho.gov" fromAddress = "rcrew@itd.idaho.gov" host = "10.1.9.67" port = "25" Properties mprops = new Properties(); mprops.setProperty("mail.transport.protocol", "smtp"); mprops.setProperty("mail.host", host); mprops.setProperty("mail.smtp.port", port); Session lSession = Session.getDefaultInstance(mprops,null); ; MimeMessage msg = new MimeMessage(lSession); //tokenize out the recipients in case they came in as a list StringTokenizer tok = new StringTokenizer(toAddress, ","); ArrayList emailTos = new ArrayList(); while(tok.hasMoreElements()){ emailTos.add(new InetAddress(tok.nextElement()).toString()); } InetAddress[] to = new InetAddress[emailTos.size()]; to = (InetAddress[]) emailTos.toArray(to); msg.setRecipients(MimeMessage.RecipientType.TO,to); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> InternetAddress fromAddr = new InternetAddress(fromAddress); msg.setFrom(fromAddr); msg.setFrom(new InternetAddress(fromAddress)); msg.setSubject(subject); msg.setText(message) Transport transporter = Session.getTransport("smtp"), transporter.connect(); transporter.send(msg); </pre>
49	MMS Model CP test2	<pre> //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script Double futureBudget=currentBudget; if (currentBudget!=null && currentBudget>0) { if (targetScore==currentScore) return currentBudget; else if (currentScore<targetScore) { futureBudget=futureBudget*targetScore/ currentScore; } else { futureBudget=futureBudget*targetScore/ currentScore; } } return futureBudget; </pre>
48	MMS Model CP test	<pre> //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script Double futureBudget=currentBudget; if (currentBudget!=null && currentBudget>0) { if (targetScore==currentScore) return currentBudget; else if (currentScore<targetScore) { futureBudget=futureBudget*targetScore/ currentScore; } else { futureBudget=futureBudget*targetScore/ currentScore; } } return futureBudget; </pre>
47	MMS Model 1	<pre> double[] coef_arr=[1.2,1.3,1.4,1.5,2]; //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> Double futureBudget=currentBudget; if (currentBudget!=null && currentBudget>0) { if (targetScore==currentScore) return currentBudget; else if (currentScore<targetScore) { for (int i=currentScore;i<targetScore ;i++) futureBudget=futureBudget*coef_arr [i- 1]; } else { for (int i=currentScore -1 ;i>=targetScore ;i--) futureBudget=futureBudget/coef_arr [i- 1]; } } return futureBudget; </pre>
46	MMS Model from	<pre> import com.agileassetsinc.core.*; DataStore dsCoef = dl.createRetrieveDataStore("select DEFECT_SCORE, DEFECT_SCORE_ACT_COEF from los_activity_coef where defect_id=? order by DEFECT_SCORE", null, defectId); dsCoef.buildIndex("idx", "DEFECT_SCORE"); double[] coef_arr=new double[5]; for(int i=1; i<=coef_arr.length; i++) { int row = dsCoef.findByName("idx", i); if(row>0) coef_arr[i-1] = dsCoef.getItemNumber(row,"DEFECT_ SCORE_ACT_COEF"); } //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script Double futureBudget=currentBudget; if (currentBudget!=null && currentBudget>0) { if (targetScore==currentScore) return currentBudget; else if (currentScore<targetScore) { for (int i=currentScore;i<targetScore ;i++) futureBudget=futureBudget*coef_arr [i- 1]; } else { for (int i=currentScore -1 ;i>=targetScore ;i--) futureBudget=futureBudget/coef_arr [i- 1]; } } </pre>

ID	Groovy Script Name	Groovy Script Text
50	MMS Util Function 1	<pre> } } return futureBudget; double[] util_arr=[0,1,3,4,6,6]; //OWNER_NET, FUNCT_CLASS_ID columns are available in ds if necessary in groovy script //for Indidana defect_score - input is integer 1,2,3,4,5,6 if (currentScore==null) return null; else return util_arr[currentScore - 1]; import com.agileassetsinc.core.*; //////////////////////////////////// //Configuration Section //////////////////////////////////// Integer utilVal=currentScore; if (currentScore>90) utilVal=100; return utilVal; </pre>
-21	General: All Above 90 go to 100	
329	Update GHG from MWP	<pre> dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.CA_GHG_DIFF = " + " (SELECT case when sum(B.LANE_MILES) <>0 then SUM (B.CA_GHG_DIFF*B.LANE_MILES)/sum(B.LANE_MILES) else null end " + " FROM user_optimization_engine b " + " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
348	IRI Benefit from MWP	<pre> // Calculated from the standard column BENEFIT_MWP dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.BENEFIT = " + " (SELECT case when sum(B.LANE_MILES) <>0 then SUM((B.BENEFIT_ALT_1- B.BENEFIT)*B.LANE_MILES)/sum(B.LA NE_MILES) else null end " + " FROM user_optimization_engine b " + </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
340	Update MWPLaneMiles	<pre> /*dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.LANE_MILES = " + " nvl((SELECT SUM (B.LANE_MILES) " + " FROM user_optimization_engine b " + " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to),A.CA_BEN_IRI_TRAFFIC_P RES) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn);*/ dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.LANE_MILES = (select pms_lane_miles from pms_master_wp where loc_ident=a.mwp_loc_ident)" + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
320	IRI/Traffic Weighted Benefit	<pre> dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.CA_BEN_IRI_TRAFFIC_DIFF = " + " (SELECT case when sum(B.LANE_MILES) <>0 then SUM (B.CA_BEN_IRI_TRAFFIC_DIFF*B.LAN E_MILES)/sum(B.LANE_MILES) else null end " + " FROM user_optimization_engine b " + " WHERE b.user_id = a. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
317	Calculate MWP Benefit IRI TRaffic	<pre> dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.CA_BEN_IRI_TRAF_PRES_DIFF = " + " nvl((SELECT case when sum(B.LANE_MILES) <>0 then SUM (B.CA_BEN_IRI_TRAF_PRES_DIFF*B. LANE_MILES)/sum(B.LANE_MILES) else null end " + " FROM user_optimization_engine b " + " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to),A.CA_BEN_IRI_TRAF_P RES) " + </pre>
306	Update Preservation benefit for MWP	<pre> dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.CA_BENEFIT_PRES_DIFF = " + " (SELECT case when sum(B.LANE_MILES) <>0 then SUM (B.CA_BENEFIT_PRES_DIFF*B.LANE_ MILES)/sum(B.LANE_MILES) else null end " + " FROM user_optimization_engine b " + " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a. </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
316	UpdateIRI/Road User Costs for MWP	<pre> dl.executeSQL("UPDATE pms_analysis_user_master_wp a " + " SET A.CA_BEN_IRI_RUC_DIFF = " + " (SELECT case when sum(B.LANE_MILES) <>0 then SUM (B.CA_BEN_IRI_RUC_DIFF*B.LANE_M ILES)/sum(B.LANE_MILES) else null end " + " FROM user_optimization_engine b " + " WHERE b.user_id = a.user_id " + " AND b.treatment_cost_alt_1 > 0 " + " AND b.route_id = a.route_id " + " AND (b.lane_dir = a.lane_dir OR a.lane_dir = 0) " + " AND (a.lane_id = b.lane_id OR a.lane_id = 0) " + " AND (b.offset_from + b.offset_to) BETWEEN 2*a.offset_from AND 2*a.offset_to) " + " WHERE a.user_id = get_user_id () AND a.plan_year = "+yearIn); </pre>
87	OnEditQaRequested	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for traffic QA procedures */ String QA_CHANGE_STR; /* Check to see if checking or unchecking. If Unchecking column then disallow */ if (dsln.getItemObjectOriginal(1,columnIn) ==1 && newValueIn != "1") { throw new TRDIEException(1242); /*You cannot clear a QA Request directly. They must be cleared by changing the QA Status field.*/ } else if (newValueIn != "1") { /* this case occurs when the user is clearing the QA requested checkbox before they saved their data basically the user is undoing their own QA request All the following statements check to see if original data value was null, if not set back to original value, </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if it was null set it back to null.*/ if (dsIn.getItemObjectOriginal(1,"ID_QA_RE VW_STATUS_ID")!= null) { dsIn.setItemNumber(1,"ID_QA_REVW_ STATUS_ID",dsIn.getItemObjectOriginal (1,"ID_QA_REVW_STATUS_ID")); } else { dsIn.setItemNull(1,"ID_QA_REVW_STA TUS_ID"); } if (dsIn.getItemObjectOriginal(1,"ID_QA_R EQUEST_DATE")!= null) { dsIn.setItemDate(1,"ID_QA_REQUEST_ DATE",dsIn.getItemObjectOriginal(1,"ID _QA_REQUEST_DATE")); } else { dsIn.setItemNull(1,"ID_QA_REQUEST_ DATE"); } if (dsIn.getItemObjectOriginal(1,"ID_QA_R EQUEST_USER")!= null) { dsIn.setItemString(1,"ID_QA_REQUEST _USER",dsIn.getItemObjectOriginal(1,"I D_QA_REQUEST_USER")); } else { dsIn.setItemNull(1,"ID_QA_REQUEST_ USER"); } if (dsIn.getItemObjectOriginal(1,"ID_QA_R EQUEST_COMMENTS")!= null) { dsIn.setItemString(1,"ID_QA_REQUEST _COMMENTS",dsIn.getItemObjectOrigin al(1,"ID_QA_REQUEST_COMMENTS")); } else { dsIn.setItemNull(1,"ID_QA_REQUEST_ COMMENTS"); } if (dsIn.getItemObjectOriginal(1,"ID_QA_R EVIEW_DATE")!= null) { dsIn.setItemDate(1,"ID_QA_REVIEW_D ATE",dsIn.getItemObjectOriginal(1,"ID_ QA_REVIEW_DATE")); } else { dsIn.setItemNull(1,"ID_QA_REVIEW_D ATE"); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } if (dsIn.getItemObjectOriginal(1,"ID_QA_REVW_COMMENTS")!= null) { dsIn.setItemString(1,"ID_QA_REVW_COMMENTS",dsIn.getItemObjectOriginal(1,"ID_QA_REVW_COMMENTS")); } else { dsIn.setItemNull(1,"ID_QA_REVW_COMMENTS"); } if (dsIn.getItemObjectOriginal(1,"QA_CHANGE_LOG")!= null) { dsIn.setItemString(1,"QA_CHANGE_LOG",dsIn.getItemObjectOriginal(1,"QA_CHANGE_LOG")); } else { dsIn.setItemNull(1,"QA_CHANGE_LOG"); } } if (newValueln=="1"){ //IF qa is being requested if (dsIn.getItemObjectOriginal(1,"ID_QA_REVW_STATUS_ID")==null) { //IF QA Status = null then no other QA actions have occurred /* ID_QA_REVIEW=Checked ID_QA_REVW_STATUS_ID=Set to Pending ID_QA_REQUEST_DATE=Sys Insert Date ID_QA_REQUEST_USER=Sys Insert User Name ID_QA_REQUEST_COMMENTS=Require User to Enter Comments ***** HAppens on record save not item change ID_QA_REVIEW_DATE=NULL ID_QA_REVW_COMMENTS=NULL ID_QA_REVIEWER=NULL QA_CHANGE_LOG=No Action */ dsIn.setItemNumber(1,"ID_QA_REVW_STATUS_ID",1); //set to pending.. dsIn.setItemDate(1,"ID_QA_REQUEST_DATE",new DateTime()); //Set to today dsIn.setItemString(1,"ID_QA_REQUEST_USER", dl.getSession().getUserId()); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> dsIn.setItemNull(1,"ID_QA_REVIEW_DATE"); //Set to null dsIn.setItemNull(1,"ID_QA_REVIEWER"); //Set to null dsIn.setItemNull(1,"ID_QA_REVIEW_COMMENTS"); //Set to null } else { /* IF QA Status is not null then previous QA information needs to be recorded in QA Change Log Column before writing in new information */ QA_CHANGE_STR="QA Request ["+dsIn.getItemString(1,"ID_QA_REQUEST_USER")+ ":" + TRDIUtils.dateToStr(dsIn.getItemDate(1,"ID_QA_REQUEST_DATE"),"[shortdate]")+"] \n" + "Request comments: "+ dsIn.getItemString(1,"ID_QA_REQUEST_COMMENTS") + "\n" + "QA Review ["+dsIn.getItemString(1,"ID_QA_REVIEWER")+ ":" + TRDIUtils.dateToStr(dsIn.getItemDate(1,"ID_QA_REVIEW_DATE"),"[shortdate]")+"] \n" + "QA Review comments: "+ dsIn.getItemString(1,"ID_QA_REVIEW_COMMENTS") + "\n" if (dsIn.getItemString(1,"QA_CHANGE_LOG")!=null) { QA_CHANGE_STR=QA_CHANGE_STR+"\n"+dsIn.getItemString(1,"QA_CHANGE_LOG"); } dsIn.setItemString(1,"QA_CHANGE_LOG", QA_CHANGE_STR); //Copy all previous information to the log column dsIn.setItemNumber(1,"ID_QA_REVIEW_STATUS_ID",1); //set to pending.. dsIn.setItemDate(1,"ID_QA_REQUEST_DATE",new DateTime()); //Set to today dsIn.setItemNull(1,"ID_QA_REQUEST_COMMENTS"); // Set to null dsIn.setItemString(1,"ID_QA_REQUEST_USER", dl.getSession().getUserId()); //Current user </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre>ID_QA_REVIEW_DATE"); //Set to null dsIn.setItemNull(1,"ID_QA_REVIEW_COMMENTS"); //Set to null }</pre>
258	Set Layer 99 for Milling Material	<pre>import com.agileassetsinc.core.*; /*check if layer is set to 99 for milling*/ if(newValueIn == "0") { /* Material Code 0 is Milling layer*/ dsIn.setItemNumber(1,"LAYER_ID",99); } /*Set layer id to null if material is set to non-milling and current layer is 99*/ if(newValueIn != "0" && dsIn.getItemNumber(1,"LAYER_ID")==99) { /* Material Code 0 is Milling layer*/ dsIn.setItemNull(1,"LAYER_ID"); }</pre>
259	Set Milling Material when	<pre>import com.agileassetsinc.core.*; /*check if layer is set to 99 for milling*/ if(newValueIn == "99") { /* Material Code 0 is Milling layer*/ dsIn.setItemNumber(1,"MTRL_CODE_ID",0); } if(newValueIn != "99" && dsIn.getItemNumber(1,"MTRL_CODE_ID")==0) { /* Material Code 0 is Milling layer*/ dsIn.setItemNull(1,"MTRL_CODE_ID"); }</pre>
240	assign_t_number	<pre>import com.agileassetsinc.core.*; Integer acc = TRDIUtils.strToDouble(newValueIn); String sql; sql = "select eq_t_numbers_name as eq_name from setup_eq_t_numbers where eq_t_numbers_id =" + acc ; DataStore ds = dl.createDataStore(sql); ds.retrieve(); dsIn.setItemString(1,"EQUIPMENT_NAME",ds.getItemString(1,"EQ_NAME"));</pre>

ID	Groovy Script Name	Groovy Script Text
229	Change WO Dates	<pre> import com.agileassetsinc.core.* import com.agileassetsinc.main.WoTimeInfo import com.agileassetsinc.main.WoScheduler; if(columnIn=="START_DATE") { dsIn.setItemDate(1, "START_DATE", TRDIUtils.strToDate(newValueIn)); } else if(columnIn=="END_DATE") { dsIn.setItemDate(1, "END_DATE", TRDIUtils.strToDate(newValueIn)); dsIn.setItemNumber(1, "DURATION", - 1); } else if(columnIn=="DURATION") { dsIn.setItemNumber(1, "DURATION", TRDIUtils.strToint(newValueIn)); } else if(columnIn=="START_HOUR") { dsIn.setItemNumber(1, "START_HOUR", TRDIUtils.strToint(newValueIn)); } else if(columnIn=="END_HOUR") { dsIn.setItemNumber(1, "END_HOUR", TRDIUtils.strToint(newValueIn)); dsIn.setItemNumber(1, "DURATION", - 1); } else if(columnIn=="CALENDAR_ID") { dsIn.setItemNumber(1, "CALENDAR_ID", TRDIUtils.strToint(newValueIn)); } WoTimeInfo woTimeInfo = new WoTimeInfo(); if(dsIn.getItemDate(1,"START_DATE") != null) woTimeInfo.setStartDate(dsIn.getItemD ate(1,"START_DATE")); if(dsIn.getItemInteger(1,"START_HOUR ") != null) woTimeInfo.setStartHour(dsIn.getItemInt eger(1,"START_HOUR")); if(dsIn.getItemDate(1,"END_DATE") != null) woTimeInfo.setEndDate(dsIn.getItemDat e(1,"END_DATE")); if(dsIn.getItemInteger(1,"END_HOUR") != null) woTimeInfo.setEndHour(dsIn.getItemInt eger(1,"END_HOUR")); if(dsIn.getItemInteger(1,"DURATION") != null) woTimeInfo.setDuration(dsIn.getItemInte ger(1,"DURATION")); if(dsIn.getItemInteger(1,"CALENDAR_ID ") != null) </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (1,"CALENDAR_ID")); woTimeInfo = WoScheduler.calcWoDates(dl, woTimeInfo); if(woTimeInfo.getStartDate() == null && woTimeInfo.getEndDate() == null) { throw new TRDException(1040); } dsln.setItemDate(1, "START_DATE", woTimeInfo.getStartDate()); dsln.setItemNumber(1, "START_HOUR", woTimeInfo.getStartHour()); dsln.setItemDate(1, "END_DATE", woTimeInfo.getEndDate()); dsln.setItemNumber(1, "END_HOUR", woTimeInfo.getEndHour()); dsln.setItemNumber(1, "DURATION", woTimeInfo.getDuration()); //return dsln.getItemString(1, "START_DATE")+ "\n"+ // dsln.getItemString(1, "START_HOUR")+ "\n"+ // dsln.getItemString(1, "END_DATE")+ "\n"+ // dsln.getItemString(1, "END_HOUR")+ "\n"+ // dsln.getItemString(1, "DURATION")+ "\n"+ // dsln.getItemString(1, "CALENDAR_ID"); </pre>
-23	General:	<pre> import com.agileassetsinc.core.*; /*Calculate and Set Length on edit of offsets*/ String dir_code; String otherOffsetColumn; double length_new; Integer locRefId = dl.getSession().getLocRefId(); sql_str="select column_id,column_id_to from SETUP_LOC_REF_COLUMN a" + " WHERE loc_ref_column_type_id=2" + " AND a.LOC_REF_ID = "+ locRefId ; DataStore ds = dl.createRetrieveDataStore(sql_str); switch (columnIn) { case ds.getItemString(1,"COLUMN_ID"): otherOffsetColumn=ds.getItemString(1," COLUMN_ID_TO"); </pre>

ID	Groovy Script Name	Groovy Script Text
-50	General: Retrieve Budget Category	<pre> break; case ds.getItemString(1,"COLUMN_ID_TO"): otherOffsetColumn=ds.getItemString(1," COLUMN_ID"); break; } Double offset_other = dsIn.getItemNumber(1, otherOffsetColumn); /*Only process if both offset from and to are not null*/ if(newValueIn != ""&& offset_other !=null) { /*Calculate Length*/ length_new = Math.abs(offset_other - TRDIUtils.strTodouble(newValueIn)); dsIn.setItemNumber(1,"LENGTH", length_new); } import com.agileassetsinc.core.*; String sql_str; sql_str="select cat.PMS_BUDGET_CAT_ID as BUDGET_CAT " + "from PMS_BUDGET_CAT cat, SYSTEM_CALC_EXPRESSIONS calc, "+ "MS_TREATMENT tr where tr.BUDGETC_EXPRESSION_ID = calc.EXPRESSION_ID " + "and calc.EXPRESSION_TYPE_ID = 4 " + "and TO_NUMBER(calc.EXPRESSION_TEX T) = cat.PMS_BUDGET_CAT_ID " + "and tr.PMS_TREATMENT_ID = " + newValueIn; DataStore ds = dl.createRetrieveDataStore(sql_str); if(ds.rowCount()==0) { throw new TRDIException("No Budget Category Defined for this Treatment"); } dsIn.setItemNumber(1,"PMS_BUDGET_ CAT_ID", ds.getItemNumber(1,"BUDGET_CAT")); </pre>

ID	Groovy Script Name	Groovy Script Text
-49	General: POSITIVE_AMOUNT	<pre> import com.agileassetsinc.core.TRDIException; if(Double.valueOf(newValueIn) < 0) { throw new TRDIException(237); } else { dsIn.setItemNumber(1, columnIn, Double.valueOf(newValueIn)); } return newValueIn; </pre>
-48	General: not_earlier_than_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate(newValueIn); dtIn = dtIn.round(); today = today.round(); if(today.isAfter(dtIn)) { throw new TRDIException(1362); } } return newValueIn; </pre>
-47	General:	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate(newValueIn); DateTime month_from_today; dtIn = dtIn.round(); today = today.round(); String year = today.getYear().toString(); String month = ((today.getMonth()+1).toString().size()== 2)?(today.getMonth()+1).toString() : "0"+(today.getMonth()+1).toString(); String day = (today.getDate().toString().size()==2)?to day.getDate().toString() : "0"+today.getDate().toString(); if((today.getMonth()+1)==13){ month = "01"; year = (today.getYear()+1).toString(); } month_from_today = TRDIUtils.strToDate(year+month+day); if(month_from_today.isAfter(dtIn)) </pre>

ID	Groovy Script Name	Groovy Script Text
-46	General:	<pre> { throw new TRDException(1363); } } return newValueln; import com.agileassetsinc.core.TRDException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); DateTime year_from_today; dtln = dtln.round(); today = today.round(); year_from_today = today.plusDays(366);//have to add extra day if(year_from_today.isAfter(dtln)) </pre>
-45	General:not_after_today	<pre> { throw new TRDException(1364); } } import com.agileassetsinc.core.TRDException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); dtln = dtln.round(); today = today.round(); if(today.isBefore(dtln)) </pre>
-44	General:	<pre> { throw new TRDException(1367); } } return newValueln; import com.agileassetsinc.core.TRDException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); DateTime month_from_today; dtln = dtln.round(); today = today.round(); String year = today.getYear().toString(); String month = ((today.getMonth() </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> +1).toString().size()==2)?(today.getMonth()+1).toString() : "0"+(today.getMonth()+1).toString(); String day = (today.getDate().toString().size()==2)?today.getDate().toString() : "0"+today.getDate().toString(); if((today.getMonth()+1)==13){ month = "01"; year = (today.getYear()+1).toString(); } month_from_today = TRDIUtils.strToDate(year+month+day); if(month_from_today.isBefore(dtln)) { throw new TRDIException(1368); } } return newValueln; </pre>
-43	General: not_after_year_from_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); DateTime year_from_today; dtln = dtln.round(); today = today.round(); year_from_today = today.plusDays(366);//have to add extra day if(year_from_today.isBefore(dtln)) { throw new TRDIException(1369); } } </pre>
-55	General: UNIT_COST_CHANGE	<pre> import com.agileassetsinc.core.TRDIException; if(dsln.getItemNumber(1, "MATERIAL_TRANSACTION_TYPE_ID ") != 5) { if(Double.valueOf(newValueln) < 0) { throw new TRDIException(237); } else { dsln.setItemNumber(1, columnln, Double.valueOf(newValueln)); } } else { dsln.setItemNumber(1, columnln, </pre>

ID	Groovy Script Name	Groovy Script Text
217	FILL_SIGN_DIMENSIONS	<pre> Double.valueOf(newValueIn)); } return newValueIn; import com.agileassetsinc.core.*; Integer acc = TRDIUtils.strToDouble(newValueIn); String sql; sql = "select ID_SIGN_CAT_CODE_NUM, ID_SIGN_CAT_CODE_WIDTH, ID_SIGN_CAT_CODE_WIDTH_UNIT, " + " ID_SIGN_CAT_CODE_HT, ID_SIGN_CAT_CODE_HT_UNIT, " + " CASE WHEN VAR_HT LIKE '%VAR%' THEN 1 ELSE 0 END AS VAR_HT, " + " CASE WHEN VAR_WD LIKE '%VAR%' THEN 1 ELSE 0 END AS VAR_WD " + " from SETUP_ID_SIGN_CAT_CODE " + " where ID_SIGN_CAT_CODE_ID = " + acc ; DataStore ds = dl.createDataStore(sql); ds.retrieve(); dsIn.setItemString(1,"VARIABLE_SIGN_ WIDTH",ds.getItemString(1,"VAR_WD")) ; dsIn.setItemString(1,"SIZE_WIDTH",ds. getItemString(1,"ID_SIGN_CAT_CODE_ WIDTH")); dsIn.setItemString(1,"SIZE_WIDTH_UNI T",ds.getItemString(1,"ID_SIGN_CAT_C ODE_WIDTH_UNIT")); dsIn.setItemString(1,"VARIABLE_SIGN_ HEIGHT",ds.getItemString(1,"VAR_HT")); dsIn.setItemString(1,"SIZE_HEIGHT",ds .getItemString(1,"ID_SIGN_CAT_CODE _HT")); dsIn.setItemString(1,"SIZE_HEIGHT_U NIT",ds.getItemString(1,"ID_SIGN_CAT _CODE_HT_UNIT")); dsIn.setItemString(1,"ID_SIGN_CAT_C ODE_NUM",ds.getItemString(1,"ID_SIG N_CAT_CODE_NUM")), </pre>
216	SET_SIGN_SUPPORT	<pre> import com.agileassetsinc.core.*; return "dw_3.SetItem(dw_3.GetRow(), 'SIGN_SUPPORT', dw_1.GetItem(dw_1.GetRow(), 'SUPPORT_ID'))"; </pre>

ID	Groovy Script Name	Groovy Script Text
220	NAME_THE_SUPPORT_id	<pre> import com.agileassetsinc.core.*; String sql = "select case when " + dsIn.getItemNumber(1,"ID_ROUTE_FR OM") + " IS NOT NULL then (select ID_SHORT_ROUTE_NAME from SETUP_ROUTE where ROUTE_ID = " + dsIn.getItemNumber(1,"ID_ROUTE_FR OM") + ") else null end as ID_SHORT_ROUTE_NAME from DUAL"; String dir = "select case when " + dsIn.getItemNumber(1,"LANE_DIR") + " IS NOT NULL then (select LANE_DIR_NAME from SETUP_LANE_DIR where LANE_DIR = " + dsIn.getItemNumber(1,"LANE_DIR") + ") else null end as LANE_DIR_NAME from DUAL"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); DataStore dsd = dl.createDataStore(dir); dsd.retrieve(); String support_name = ds.getItemString(1,"ID_SHORT_ROUTE _NAME") + " - " + dsIn.getItemNumber(1,"ID_MILEPOST_ FROM") + " - " + dsd.getItemString(1,"LANE_DIR_NAME") + " - " + newValueIn; if(dsIn.getItemNumber(1,"ID_ROUTE_F ROM") == null dsIn.getItemNumber(1,"LANE_DIR") == null dsIn.getItemNumber(1,"ID_MILEPOST_ FROM") == null) { throw new TRDException ("Please complete the Location details (Route, Milepost From and Direction) before filling in the Supports ID."); } dsIn.setItemString(1,"SUPPORT_NAME",support_name); </pre>
219	SIGN_WIDTH_CHECK	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"VARIABLE_SI GN_WIDTH") == 0) { throw new TRDException ("Cannot change the width on this sign. The width is not variable.") } </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dsIn.getItemNumber(1,"VARIABLE_SIGN_WIDTH") == 1) { dsIn.setItemNumber(1,"SIZE_WIDTH_UNIT",44) } </pre>
218	SIGN_HEIGHT_CHECK	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"VARIABLE_SIGN_HEIGHT") == 0) { throw new TRDIException ("Cannot change the height on this sign. The height is not variable.") } if(dsIn.getItemNumber(1,"VARIABLE_SIGN_HEIGHT") == 1) { dsIn.setItemNumber(1,"SIZE_HEIGHT_UNIT",44) } </pre>
194	SMS	<pre> import com.agileassetsinc.core.*; MailApi.send("8067867211@txt.att.net", "Test SMS", "This is a test to check whether a SMS can be sent out of AgileAssets application.", "SMS-Service@AgileAssets.com"); </pre>
192	not_after_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate(newValueIn); dtIn = dtIn.round(); today = today.round(); if(today.isBefore(dtIn)) { throw new TRDIException(1367); } } return newValueIn; </pre>
191	not_after_month_from_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (newValueIn); DateTime month_from_today; dtIn = dtIn.round(); today = today.round(); String year = today.getYear().toString(); String month = ((today.getMonth()+1).toString().size()== 2)?(today.getMonth()+1).toString() : "0"+(today.getMonth()+1).toString(); String day = (today.getDate().toString().size()==2)?to day.getDate().toString() : "0"+today.getDate().toString(); if((today.getMonth()+1)==13){ month = "01"; year = (today.getYear()+1).toString(); } month_from_today = TRDIUtils.strToDate(year+month+day); if(month_from_today.isBefore(dtIn)) { throw new TRDIException(1368); } } return newValueIn; </pre>
190	not_after_year_from_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate(newValueIn); DateTime year_from_today; dtIn = dtIn.round(); today = today.round(); year_from_today = today.plusDays(366);//have to add extra day if(year_from_today.isBefore(dtIn)) { throw new TRDIException(1369); } } </pre>
189	not_earlier_than_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueIn.isEmpty()){ DateTime today = new DateTime(); DateTime dtIn = TRDIUtils.strToDate(newValueIn); </pre>

ID	Groovy Script Name	Groovy Script Text
188	not_ealier_than_month_from_today	<pre> dtln = dtln.round(); today = today.round(); if(today.isAfter(dtln)) { throw new TRDIException(1362); } } return newValueln; import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); DateTime month_from_today; dtln = dtln.round(); today = today.round(); String year = today.getYear().toString(); String month = ((today.getMonth()+1).toString().size()== 2)?(today.getMonth()+1).toString() : "0"+(today.getMonth()+1).toString(); String day = (today.getDate().toString().size()==2)?to day.getDate().toString() : "0"+today.getDate().toString(); if((today.getMonth()+1)==13){ month = "01"; year = (today.getYear()+1).toString(); } month_from_today = TRDIUtils.strToDate(year+month+day); if(month_from_today.isAfter(dtln)) { throw new TRDIException(1363); } } return newValueln; </pre>
187	not_ealier_than_year_from_today	<pre> import com.agileassetsinc.core.TRDIException import com.agileassetsinc.core.DateTime; import com.agileassetsinc.core.TRDIUtils; if(!newValueln.isEmpty()){ DateTime today = new DateTime(); DateTime dtln = TRDIUtils.strToDate(newValueln); DateTime year_from_today; dtln = dtln.round(); today = today.round(); year_from_today = today.plusDays(366);//have to add extra day </pre>

ID	Groovy Script Name	Groovy Script Text
224	NAME_THE_POLE	<pre> { throw new TRDException(1364); } } return newValueIn; import com.agileassetsinc.core.*; String sql; sql = "select substr(b.support_name,0,instr(b.support _name,'-',1,3)-2) as SUPPORT_NAME from SUPPORT_INVENTORY b " + " where SUPPORT_ID = " + dsIn.getItemNumber(1,"SUPPORT_ID") ; DataStore ds = dl.createDataStore(sql); ds.retrieve(); String pos = "select DECODE("+newValueIn+",1,'A', " + " 2,'B', " + " 3,'C', " + " 4,'D') as POLE_POSITION from dual"; DataStore dsp = dl.createDataStore(pos); dsp.retrieve(); String pole_name = ds.getItemString(1,"SUPPORT_NAME") + " - " + dsp.getItemString(1,"POLE_POSITION") dsIn.setItemString(1,"ID_SUPPORT_PO LE_NAME",pole_name); </pre>
223	NAME_THE_SUPPORT_mp	<pre> import com.agileassetsinc.core.*; //When Changing Value of from milepost lookup the segment code //Code works on edit of ID_MILEPOST_FROM and ID_MILEPOST_TO columns Integer locRefId = dl.getSession().getLocRefId(); // Only operate is not in Basic Location Reference int route_from; String sql_str; boolean from_edit; DataStore ds; if(columnIn=="ID_MILEPOST_FROM") { from_edit=true; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } else { from_edit=false; } route_from = dsIn.getItemInteger(1,"ID_ROUTE_FROM"); if (!LRS.getInstance().getLocRef(locRefId) .isBasic()) { if(newValueln != "" && route_from != null) { if (from_edit) { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM<= "+ newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and source_table='LOC_REF_42'"; } else { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM< "+ newValueln + " and a.ID_MILEPOST_TO>= "+ newValueln + " and source_table='LOC_REF_42'"; } ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0) { throw new TRDIException("The milepost "+newValueln+ " was not found on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)); } if(ds.getItemInteger(1,"CNT")>1) { if(from_edit) { dsIn.setItemString(1,"ID_SEGMENT_FROM", ""); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> } dl.getSession().setWarningString("The milepost "+newValueln+ " occurs more than once on route "+dsln.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)+ ", you must specify the desired segment code."); } else { if(from_edit) { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<=" + newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and sourse_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsln.setItemString(1,"ID_SEGMENT_FR OM",ds.getItemString(1,"ID_SEGMENT _FROM")); } else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<" + newValueln + " and a.ID_MILEPOST_TO>=" + newValueln + " and sourse_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsln.setItemString(1,"ID_SEGMENT_TO ",ds.getItemString(1,"ID_SEGMENT_FR OM")); } } } } String sql = "select case when " + dsln.getItemNumber(1,"ID_ROUTE_FR OM") + " IS NOT NULL then (select ID_SHORT_ROUTE_NAME from SETUP_ROUTE where ROUTE_ID </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> = " + dsln.getItemNumber(1,"ID_ROUTE_FR OM") + ") else null end as ID_SHORT_ROUTE_NAME from DUAL"; String dir = "select case when " + dsln.getItemNumber(1,"LANE_DIR") + " IS NOT NULL then (select LANE_DIR_NAME from SETUP_LANE_DIR where LANE_DIR = " + dsln.getItemNumber(1,"LANE_DIR") + ") else null end as LANE_DIR_NAME from DUAL"; DataStore dsl = dl.createDataStore(sql); dsl.retrieve(); DataStore dsd = dl.createDataStore(dir); dsd.retrieve(); String support_name = dsl.getItemString(1,"ID_SHORT_ROUT E_NAME") + " - " + newValueln + " - " + dsd.getItemString(1,"LANE_DIR_NAME") + " - " + dsln.getItemString(1,"TEMP_SUPPORT _ID"); dsln.setItemString(1,"SUPPORT_NAME",support_name); </pre>
222	NAME_THE_SUPPORT_dir	<pre> import com.agileassetsinc.core.*; String sql = "select case when " + dsln.getItemNumber(1,"ID_ROUTE_FR OM") + " IS NOT NULL then (select ID_SHORT_ROUTE_NAME from SETUP_ROUTE where ROUTE_ID = " + dsln.getItemNumber(1,"ID_ROUTE_FR OM") + ") else null end as ID_SHORT_ROUTE_NAME from DUAL"; String dir = "select case when " + newValueln + " IS NOT NULL then (select LANE_DIR_NAME from SETUP_LANE_DIR where LANE_DIR = " + newValueln + ") else null end as LANE_DIR_NAME from DUAL"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); DataStore dsd = dl.createDataStore(dir); dsd.retrieve(); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre>String support_name = ds.getItemString(1,"ID_SHORT_ROUTE_NAME") + " - " + dsIn.getItemNumber(1,"ID_MILEPOST_FROM") + " - " + dsd.getItemString(1,"LANE_DIR_NAME") + " - " + dsIn.getItemString(1,"TEMP_SUPPORT_ID");</pre>
221	NAME_THE_SUPPORT_route	<pre>dsIn.setItemString(1,"SUPPORT_NAME",support_name); import com.agileassetsinc.core.*; String sql = "select case when " + newValueln + " IS NOT NULL then (select ID_SHORT_ROUTE_NAME from SETUP_ROUTE where ROUTE_ID = " + newValueln + ") else null end as ID_SHORT_ROUTE_NAME from DUAL"; String dir = "select case when " + dsIn.getItemNumber(1,"LANE_DIR") + " IS NOT NULL then (select LANE_DIR_NAME from SETUP_LANE_DIR where LANE_DIR = " + dsIn.getItemNumber(1,"LANE_DIR") + ") else null end as LANE_DIR_NAME from DUAL"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); DataStore dsd = dl.createDataStore(dir); dsd.retrieve(); String support_name = ds.getItemString(1,"ID_SHORT_ROUTE_NAME") + " - " + dsIn.getItemNumber(1,"ID_MILEPOST_FROM") + " - " + dsd.getItemString(1,"LANE_DIR_NAME") + " - " + dsIn.getItemString(1,"TEMP_SUPPORT_ID");</pre>
193	MMS	<pre>dsIn.setItemString(1,"SUPPORT_NAME",support_name); import com.agileassetsinc.core.*; DataStore ds = dl.createRetrieveDataStore("select * from ams_id.comments_det where comment_id = " + dsIn.getItemInteger(1,"COMMENT_ID"))</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(dsIn.getItemNumber(1,"COMMENT_I D") == null) { throw new TRDException("Please upload a picture to send as MMS."); } File tempDir = SystemSetup.getInstance().getTempSys Dir(); String fileName = ds.getItemString(1,"COMMENT_NAME"); byte[] data = ds.getItemBlob(1,"COMMENT_DATA"). getBytes(); File tempFile = new File(fileName, tempDir); FileOutputStream fos = new FileOutputStream(tempFile); fos.write(data); MailApi.send("8067867211@mms.att.ne t", "Test MMS", "This is a test to check whether a MMS can be sent out of AgileAssets application.", "MMS- Service@AgileAssets.com", SystemSetup.getInstance().getMailHost(), SystemSetup.getInstance().getMailUser name(), SystemSetup.getInstance().getMailPass import com.agileassetsinc.core.* /*This script processes the following rules for ITD PATCHES */ /* Check to see if checking or unchecking. If Unchecking column then clear */ if (newValueIn=="1"){ dsIn.setItemDate(1,"d_DATE",new DateTime()); dsIn.setItemString(1,"d_USER", dl.getSession().getUserId()); } else { dsIn.setItemNull(1,"d_DATE"); dsIn.setItemString(1,"d_USER",""); } </pre>
182	ITD_DEV_PATCH_CHANGE	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for ITD PATCHES */ /* Check to see if checking or unchecking. If Unchecking column then clear */ if (newValueIn=="1"){ dsIn.setItemDate(1,"d_DATE",new DateTime()); dsIn.setItemString(1,"d_USER", dl.getSession().getUserId()); } else { dsIn.setItemNull(1,"d_DATE"); dsIn.setItemString(1,"d_USER",""); } </pre>
180	Update_Pavement_Type	<pre> import com.agileassetsinc.core.*; /*When Changing Value of HPMS PAVEMENT set TAMS pavement type </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> */ String sql_str; DataStore ds; if(newValueIn!="") { sql_str="select WC_ID as TAMS_WC_ID from SETUP_ID_ASC_PAV_TYP where ID_ASC_PAV_TYP_ID="+ newValueIn; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemNumber(1,"WC_ID",ds.getItemInteger(1,"TAMS_WC_ID")); } </pre>
179	ITD_WAR_PATCH_CHANGE	<pre> import com.agileassetsinc.core.* if (newValueIn=="1") { dsIn.setItemString(1,"ITD_PATCH_STA </pre>
174	ITD_TEST_PATCH_CHANGE	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for ITD PATCHES */ /* Check to see if checking or unchecking. If Unchecking column then clear */ if (newValueIn=="1"){ dsIn.setItemDate(1,"T_DATE",new DateTime()); dsIn.setItemString(1,"T_USER", dl.getSession().getUserId()); } else { dsIn.setItemNull(1,"T_DATE"); dsIn.setItemString(1,"T_USER",""); } </pre>
173	ITD_TRAINING_PATCH_CHANGE	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for ITD PATCHES */ /* Check to see if checking or unchecking. If Unchecking column then clear */ if (newValueIn=="1"){ dsIn.setItemDate(1,"tr_DATE",new DateTime()); dsIn.setItemString(1,"tr_USER", dl.getSession().getUserId()); } else { dsIn.setItemNull(1,"tr_DATE"); dsIn.setItemString(1,"tr_USER",""); </pre>

ID	Groovy Script Name	Groovy Script Text
		}
178	ITD_PROD_PATCH_CHANGE	<pre> import com.agileassetsinc.core.* /*This script processes the following rules for ITD PATCHES */ /* Check to see if checking or unchecking. If Unchecking column then clear */ if (newValueIn=="1"){ dsIn.setItemDate(1,"P_DATE",new DateTime()); dsIn.setItemString(1,"P_USER", dl.getSession().getUserId()); } else { dsIn.setItemNull(1,"P_DATE"); dsIn.setItemString(1,"P_USER",""); } </pre>
159	Approved Check and Set Segment	<pre> import com.agileassetsinc.core.*; /*When Changing Value of from milepost lookup the segment code Code works on edit of ID_MILEPOST_FROM and ID_MILEPOST_TO columns*/ Integer locRefId = dl.getSession().getLocRefId(); /* Only operate is not in Basic Location Reference*/ int route_from; String sql_str; boolean from_edit; DataStore ds; if(dsIn.getItemNumber(1,"APPROVED") == 1) { throw new TRDException("Cannot change Approved record"); } if(columnIn=="ID_MILEPOST_FROM") { from_edit=true; } else { from_edit=false; } route_from = dsIn.getItemInteger(1,"ID_ROUTE_FRO M"); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(newValueln != "" && route_from != null) { if (from_edit) { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM<= "+ newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and source_table='LOC_REF_42'"; } else { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM< "+ newValueln + " and a.ID_MILEPOST_TO>= "+ newValueln + " and source_table='LOC_REF_42'"; } ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0) { throw new TRDException("The milepost "+newValueln+ " was not found on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)); } if(ds.getItemInteger(1,"CNT")>1) { if(from_edit) { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); } else { dsIn.setItemString(1,"ID_SEGMENT_TO ",""); } dl.getSession().setWarningString("The milepost "+newValueln+ " occurs more than once on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)+ " , you must specify the desired segment code."); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> else { if(from_edit) { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<=" + newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_FR OM",ds.getItemString(1,"ID_SEGMENT _FROM")); } else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM< " + newValueln + " and a.ID_MILEPOST_TO>= "+ newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_TO ",ds.getItemString(1,"ID_SEGMENT_FR OM")); } } </pre>
158	INV_ELEM_APPROVED_CHECK	<pre> } import com.agileassetsinc.core.* if(dsIn.getItemNumber(1,"APPROVED") == 1){ throw new TRDException("Cannot change Approved record"); </pre>
157	Lane_Dir_default_on_Inv_Element	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"APPROVED") == 1) { throw new TRDException("Cannot change </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> //Integer woid = dsIn.getItemNumber(1,"WORK_ORDER_ID"); // String sql; // sql = "select a.parent_id from setup_activity a, work_orders b " + //" where a.activity_id = b.activity_id " + // " and b.work_order_id = " + woid ; //DataStore ds = dl.createDataStore(sql); //ds.retrieve(); // if(ds.getItemNumber(1,"PARENT_ID") != 8) { if(dsIn.getItemNumber(1,"LANE_ID") == null) { dsIn.setItemNumber(1,"LANE_ID",0); } if(dsIn.getItemNumber(1,"LANE_DIR") == null) { dsIn.setItemNumber(1,"LANE_DIR",0); } // } </pre>
156	Lane_Dir_default on WR	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"WR_APPROV ED") == 1) { throw new TRDException("Cannot change Approved record"); } if(dsIn.getItemNumber(1,"LANE_ID") == null) { dsIn.setItemNumber(1,"LANE_ID",0); } if(dsIn.getItemNumber(1,"LANE_DIR") == null) { dsIn.setItemNumber(1,"LANE_DIR",0); } </pre>

ID	Groovy Script Name	Groovy Script Text
155	WO_LOC_APPROVE_CHECK	<pre>import com.agileassetsinc.core.* if(dsIn.getItemNumber(1,"WR_APPROVED") == 1){ throw new TRDIEException("Cannot change Approved record"); }</pre>
153	fill_district_on_wr	<pre>import com.agileassetsinc.core.*; String sql; sql = "SELECT b.OWNER_ID " + " FROM SETUP_OWNER b " + " WHERE b.OWNER_ID IN (SELECT c.OWNER_ID FROM SETUP_OWNER c CONNECT BY PRIOR c.PARENT_ID = c.OWNER_ID START WITH c.OWNER_ID = GET_OWNER_ID()) " + " AND b.LEVEL_ID = CASE WHEN (SELECT d.LEVEL_ID FROM SETUP_OWNER d WHERE d.OWNER_ID = GET_OWNER_ID()) < 5 " + " THEN (SELECT d.LEVEL_ID FROM SETUP_OWNER d WHERE d.OWNER_ID = GET_OWNER_ID()) " + " ELSE 4 " + " END"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); dsIn.setItemString(1,"DISTRICT_ID",ds. getItemString(1,"OWNER_ID")); if(dsIn.getItemNumber(1,"WR_APPROVED") == 1){ throw new TRDIEException("Cannot change Approved record"); }</pre>
150	set_budget_category on edit	<pre>import com.agileassetsinc.core.*; //This script will set the value of the budget category column when editing the treatment String sql_str; DataStore ds; if(newValueIn==null) { dsIn.setItemNull(1,"PMS_BUDGET_CAT_ID"); } else { sql_str= "select to_number(b.expression_text) as PMS_BUDGET_CAT_ID from ms_treatment a,</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> system_calc_expressions b " + "where A.BUDGETC_EXPRESSION_ID=B.EXP RESSION_ID and A.PMS_TREATMENT_ID="+newValueIn ; ds = dl.createRetrieveDataStore(sql_str); if(dsIn.rowCount()>=1) { dsIn.setItemNumber(1, "PMS_BUDGET_CAT_ID",ds.getItemNu mber(1,"PMS_BUDGET_CAT_ID")); } } </pre>
148	User_Check	<pre> import com.agileassetsinc.core.*; String sql; sql = "SELECT USER_ID FROM SETUP_USER_TABLE WHERE USER_ID = Get_User_Id()"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); if(dsIn.getItemString(1,"USER_UPDATE ") != ds.getItemString(1,"USER_ID")) { throw new TRDException("Cannot change record inserted by another user"); } </pre>
146	Target_stock_check_on_plan_amt	<pre> import com.agileassetsinc.core.*; Integer Activity = dsIn.getItemNumber(1,"ACTIVITY_ID"); Integer Stock = dsIn.getItemNumber(1,"MIX_TARGET_ STOCK_ID"); String sql; sql = "SELECT CASE WHEN " + Activity + " IN (SELECT ACTIVITY_ID FROM MATERIAL_MIX_ACTIVITY_MASTER) THEN 1 ELSE 0 END AS ActCheck FROM DUAL"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); if(Stock==null && ds.getItemNumber(1,"ActCheck")==1) { throw new TRDException("Enter Mix Target Stock ID"); } </pre>

ID	Groovy Script Name	Groovy Script Text
145	SET_WO_LOC_DATE	<pre> import com.agileassetsinc.core.*; Integer woid = dsIn.getItemNumber(1,"WORK_ORDER_ID"); String sql; sql = "SELECT START_DATE FROM WORK_ORDERS WHERE WORK_ORDER_ID = " + woid; DataStore ds = dl.createDataStore(sql); ds.retrieve(); </pre>
147	Target_stock_check_on_complete	<pre> dsIn.setItemString(1,"DATE_WORK",ds. getItemString(1,"START_DATE")); import com.agileassetsinc.core.*; Integer Activity = dsIn.getItemNumber(1,"ACTIVITY_ID"); Integer Stock = dsIn.getItemNumber(1,"MIX_TARGET_STOCK_ID"); String sql; sql = "SELECT CASE WHEN " + Activity + " IN (SELECT ACTIVITY_ID FROM MATERIAL_MIX_ACTIVITY_MASTER) THEN 1 ELSE 0 END AS ActCheck FROM DUAL"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); if(Stock==null && ds.getItemNumber(1,"ActCheck")==1) { throw new TRDException("Cannot Complete the Work Order. The Mix Target Stock ID cannot be left blank for a mixing activity WO."); } </pre>
142	ON_EDIT_IS_TEMPLATE_SCENARIO	<pre> /* Script to set or clear the owner id of a scenario to null when users lock and share it so all owners can see it.*/ if (newValueIn != "1") { dsIn.setItemNumber(1,"OWNER_ID", dl.getSession().getOwnerId()); } else { dsIn.setItemNull(1,"OWNER_ID"); } </pre>
112	Set Work Code (no check for	<pre> import com.agileassetsinc.core.*; /*When Changing Value of treatment set the proper work code in table without approved column*/ String sql_str; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> DataStore ds; if(newValueIn!="") { sql_str="select pms_work_code_id as PMS_WORK_CODE_ID from ms_treatment where pms_treatment_id="+ newValueIn; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemNumber(1,"PMS_WORK_C ODE_ID",ds.getItemInteger(1,"PMS_W ORK_CODE_ID")); } </pre>
96	WR_CAT_CHECK	<pre> import com.agileassetsinc.core.*; Double CATEGORY = Double.valueOf(newValueIn); //Double REQUEST = dsIn.getItemNumber(1,"WORK_REQUE ST_TYPE_ID"); if(CATEGORY != null && dsIn.getItemNumber(1,"WORK_REQUE ST_TYPE_ID") != 2) { throw new TRDIException("To select an Emergency Category, the Work Request Type selected should be *Emergency* "); } if(dsIn.getItemNumber(1,"WR_APPROV ED") == 1){ throw new TRDIException("Cannot change Approved record"); } </pre>
95	WR_START_DATE_CHECK	<pre> import com.agileassetsinc.core.*; DateTime startDate = TRDIUtils.strToDate(newValueIn); DateTime endDate = dsIn.getItemDate(1,"WR_END_DATE"); if(startDate!=null && endDate!=null && startDate.isAfter(endDate)) { throw new TRDIException("Start Date cannot be newer than End Date"); } if(dsIn.getItemNumber(1,"WR_APPROV ED") == 1) { throw new TRDIException("Cannot change Approved record"); } </pre>

ID	Groovy Script Name	Groovy Script Text
94	WR_END_DATE_CHECK	<pre> import com.agileassetsinc.core.*; DateTime startDate = dsIn.getItemDate(1,"WR_START_DATE "); DateTime endDate = TRDIUtils.strToDate(newValueIn); if(startDate!=null && endDate!=null && startDate.isAfter(endDate)) { throw new TRDIException("End Date cannot be older than Start Date"); } </pre>
82	id from equals to script	<pre> if(dsIn.getItemNumber(1,"WR_APPROV ED") == 1) { throw new TRDIException("Cannot change import com.agileassetsinc.core.*; //Distance From Origin if(columnIn.equals("OFFSET_FROM")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"OFFSET_TO"); } else { dsIn.setItemNumber(1,"OFFSET_TO",T RDIUtils.strToDouble(newValueIn)); } } else if(columnIn.equals("OFFSET_TO")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"OFFSET_FROM"); } else { dsIn.setItemNumber(1,"OFFSET_FROM ",TRDIUtils.strToDouble(newValueIn)); } } //ID Route/Milepost/Seg else if(columnIn.equals("ID_ROUTE_FROM")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"ID_ROUTE_TO"); } else { dsIn.setItemNumber(1," </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ID_ROUTE_TO",TRDIUtils.strToDouble(newValueIn)); } } else if(columnIn.equals("ID_ROUTE_TO")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"ID_ROUTE_FROM") ; } else { dsIn.setItemNumber(1,"ID_ROUTE_FR OM",TRDIUtils.strToDouble(newValueIn)); } } else if(columnIn.equals("ID_MILEPOST_FRO M")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"ID_MILEPOST_TO") ; } else { dsIn.setItemNumber(1,"ID_MILEPOST_ TO",TRDIUtils.strToDouble(newValueIn)); } } else if(columnIn.equals("ID_MILEPOST_TO")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"ID_MILEPOST_FRO M"); } else { dsIn.setItemNumber(1,"ID_MILEPOST_ FROM",TRDIUtils.strToDouble(newValu eIn)); } } else if(columnIn.equals("ID_SEGMENT_FRO M")) { if(newValueIn==null newValueIn.equals("")) { dsIn.setItemNull(1,"ID_SEGMENT_TO") ; } else { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ID_SEGMENT_FROM"); } else { </pre>
88	onEditDateSetYear	<pre> dsIn.setItemNumber(1,"ID_SEGMENT_ FROM",TRDIUtils.strToDouble(newValu eIn)); } import com.agileassetsinc.core.* /* Sets the year value to the first four digits of the data value good for tables with column EFF_YEAR*/ if (newValueIn!=null) { dsIn.setItemNumber(1, "EFF_YEAR", TRDIUtils.strToInteger(newValueIn.subs tring(0,4))); } </pre>
78	RFP Set Date Approved	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemNumber(1,"RFP_APPRO VED")!=1 && dsIn.getItemDate(1,"RFP_DATE_APPR OVED") == null) { dsIn.setItemDate(1,"RFP_DATE_APPR OVED", new DateTime()); } </pre>
41	Multiply Index by 20	<pre> import com.agileassetsinc.core.TRDIException; if(Double.valueOf(newValueIn) < 0) { throw new TRDIException(237); } else { dsIn.setItemNumber(1, "CI_100", Double.valueOf(newValueIn)*20); } ; </pre>
75	Test warming confirm	<pre> return " if(basic_window.ShowConfirm('You should not do this. Do you really want to?')) {} else {EventReturnValue = -1;}"; </pre>
86	SET_ROUTE_ON_EDIT_SEGMENT	<pre> import com.agileassetsinc.core.*; /*When Changing Value of from or to segment check to see if other segment is on the same route if so set route to the route from entered segment is valid, if not make milepost from and to null, otherwise leave alone and set route properly for from and to) Code works on edit of ID_SEGMENT_FROM and TO columns*/ Integer locRefId = dl.getSession(). </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> getLocRefId(); /* Only operate is not in Basic Location Reference*/ String sql_str; boolean from_edit; DataStore ds; Double mp_from; Double mp_to; String seg_from; String seg_to; mp_from = dsIn.getItemInteger(1,"ID_MILEPOST_F ROM"); mp_to = dsIn.getItemInteger(1,"ID_MILEPOST_T O"); route_from=dsIn.getItemInteger(1,"ID_R OUTE_FROM"); route_to=dsIn.getItemInteger(1,"ID_RO UTE_TO"); if (!LRS.getInstance().getLocRef(locRefId) .isBasic()) { if(newValueIn != "" && mp_from != "") { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + newValueIn + " and a.ID_MILEPOST_FROM<= " + mp_from + " and a.ID_MILEPOST_TO> " + mp_from + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0 ds.get ItemInteger(1,"CNT")>1) { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); dsIn.setItemString(1,"ID_MILEPOST_FR OM",""); } else {sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ newValueIn + " and a.ID_MILEPOST_FROM<= " + mp_from + " and a.ID_MILEPOST_TO> " + </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> (sql_str); dsIn.setItemString(1,"ID_SEGMENT_FROM",ds.getItemString(1,"ID_SEGMENT_FROM")); } } if(newValueIn != "" && mp_to != "") { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM="+ newValueIn + " and a.ID_MILEPOST_FROM<= "+ mp_to + " and a.ID_MILEPOST_TO> "+ mp_to + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0 ds.getItemInteger(1,"CNT")>1) { dsIn.setItemString(1,"ID_SEGMENT_TO", ""); dsIn.setItemString(1,"ID_MILEPOST_TO", ""); } else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ newValueIn + " and a.ID_MILEPOST_FROM<= "+ mp_to + " and a.ID_MILEPOST_TO> "+ mp_to + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_TO",ds.getItemString(1,"ID_SEGMENT_FROM")); } } } import com.agileassetsinc.core.*; /*When Changing Value of from milepost lookup the segment code Code works on edit of ID_MILEPOST_FROM and ID_MILEPOST_TO columns it sets the </pre>
85	Set Segment with From=to	

ID	Groovy Script Name	Groovy Script Text
		<pre> milepost columns equal and sets value of segment code if possible*/ Integer locRefId = dl.getSession().getLocRefId(); /* Only operate is not in Basic Location Reference*/ int route_from; String sql_str; boolean from_edit; DataStore ds; if(columnIn=="ID_MILEPOST_FROM") { from_edit=true; } else { from_edit=false; } route_from = dsIn.getItemInteger(1,"ID_ROUTE_FRO M"); if (!LRS.getInstance().getLocRef(locRefId) .isBasic()) { if(newValueIn != "" && route_from != null) { if (from_edit) { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM<=" + newValueIn + " and a.ID_MILEPOST_TO> "+ newValueIn + " and source_table='LOC_REF_42'"; } else { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM< "+ newValueIn + " and a.ID_MILEPOST_TO>=" + newValueIn + " and source_table='LOC_REF_42'"; } ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0) { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> found on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)); } if(ds.getItemInteger(1,"CNT")>1) { if(from_edit) { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); dsIn.setItemString(1,"ID_SEGMENT_TO ",""); dsIn.setItemString(1,"ID_MILEPOST_T O",""); } else { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); dsIn.setItemString(1,"ID_SEGMENT_TO ",""); dsIn.setItemNull(1,"ID_MILEPOST_FRO M",""); } dl.getSession().setWarningString("The milepost "+newValueln+ " occurs more than once on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)+ ", you must specify the desired segment code."); } else { if(from_edit) { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<=" + newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_FR OM",ds.getItemString(1,"ID_SEGMENT _FROM")); </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<" + newValueln + " and a.ID_MILEPOST_TO>=" + newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_FR OM",ds.getItemString(1,"ID_SEGMENT _FROM")); dsIn.setItemString(1,"ID_SEGMENT_TO ",ds.getItemString(1,"ID_SEGMENT_FR OM")); dsIn.setItemNumber(1,"ID_MILEPOST_ FROM",TRDIUtils.strTodouble(newValu eln)); } } </pre>
105	Approved updates on user and date	<pre> } import com.agileassetsinc.core.*; /*This script sets the approved user and date when someone edits the approved column from null or 0 to a value of 1 */ /* Check to see if checking or unchecking. If Unchecking column then disallow */ if (TRDIUtils.strToInteger(newValueln)==1) { dsIn.setItemDate(1,"ID_APPROV_DATE ",new DateTime()); dsIn.setItemString(1,"ID_APPROV_USE R", dl.getSession().getUserId()); } else if (TRDIUtils.strToInteger(newValueln)==0) { dsIn.setItemNull(1,"ID_APPROV_DATE"); dsIn.setItemNull(1,"ID_APPROV_USER "); } </pre>

ID	Groovy Script Name	Groovy Script Text
104	fill_acc_strip_info	<pre> import com.agileassetsinc.core.*; Integer acc = TRDIUtils.strToDouble(newValueIn); String sql; sql = "select ADV_PROG_CD, ADV_PHASE_CD, ADV_ACTV_CD, " + " ADV_TASK_CD, ADV_FY " + " from SETUP_ID_ADV_VALID_ACC_CODE " + "where ID_ADV_VALID_ACC_CODE_ID = " + acc ; String own; own = "select EXT_ADV_UNIT_CD from SETUP_OWNER where OWNER_ID = Get_Owner_Id()"; DataStore ds = dl.createDataStore(sql); ds.retrieve(); DataStore dso = dl.createDataStore(own); dso.retrieve(); dsIn.setItemString(1,"ADV_PROG_CD", ds.getItemString(1,"ADV_PROG_CD")); dsIn.setItemString(1,"ADV_PHASE_CD", ds.getItemString(1,"ADV_PHASE_CD")); dsIn.setItemString(1,"ADV_ACTV_CD",d s.getItemString(1,"ADV_ACTV_CD")); dsIn.setItemString(1,"ADV_TASK_CD",d s.getItemString(1,"ADV_TASK_CD")); dsIn.setItemString(1,"ADV_FY",ds.getIte mString(1,"ADV_FY")); dsIn.setItemString(1,"EXT_ADV_UNIT_ CD",dso.getItemString(1,"EXT_ADV_UN IT_CD")); </pre>
109	setWorkCode on Edit Treatment	<pre> import com.agileassetsinc.core.*; /*When Changing Value of treatment set the proper work code intended for use in contracts window where there is an approved column*/ String sql_str; DataStore ds; if(newValueIn!="") && dsIn.getItemInteger(1,"APPROVED")!=1) { sql_str="select pms_work_code_id as </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> ms_treatment where pms_treatment_id="+ newValueln; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemNumber(1,"PMS_WORK_CODE_ID",ds.getItemInteger(1,"PMS_W ORK_CODE_ID")); } </pre>
108	OnEditSubCompleteDateSetYear	<pre> import com.agileassetsinc.core.* /* Sets the year value to the first four digits of the data value good for tables with column EFF_YEAR*/ if (newValueln!=" " && dsIn.getItemInteger(1,"APPROVED") !=1) { dsIn.setItemNumber(1, "YEAR_COMPLETION", TRDIUtils.strToInteger(newValueln.subs tring(0,4))); } if (newValueln==" " && dsIn.getItemInteger(1,"APPROVED") !=1) { dsIn.setItemNull(1, "YEAR_COMPLETION"); } </pre>
107	Reviewed updates on user and date	<pre> import com.agileassetsinc.core.*; /*This script sets the approved user and date when someone edits the reviewed column from null or 0 to a value of 1 */ /* Check to see if checking or unchecking. If Unchecking column then disallow */ if ((dsIn.getItemObjectOriginal(1,columnIn) ==0 dsIn.getItemObjectOriginal(1,columnIn)= =null) &&TRDIUtils.strToInteger(newValueln)= =1) { dsIn.setItemDate(1,"ID_REVIEW_DATE", new DateTime()); dsIn.setItemString(1,"ID_REVIEWER", dl.getSession().getUserId()); } else if (TRDIUtils.strToInteger(newValueln)==0) { dsIn.setItemNull(1,"ID_REVIEW_DATE"); } </pre>
106	DISALLOW_KPI_EDIT_IF RECORD IS	<pre> import com.agileassetsinc.core.*; if(dsIn.getItemInteger(1,"ID_KPI_STATS _LOCK")==1) { </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> throw new TRDIEException("This record is locked and cannot be edited. It must be unlocked first."); // You cannot set the status of a previously reviewed record to null. </pre>
99	WR_APPROVE_CHECK	<pre> } import com.agileassetsinc.core.* </pre>
24	POSITIVE_AMOUNT	<pre> if(dslIn.getItemNumber(1,"WR_APPROV ED")==1){ throw new TRDIEException("Cannot change Approved record"); import com.agileassetsinc.core.TRDIEException; if(Double.valueOf(newValueIn) < 0) { throw new TRDIEException(237); } else { dslIn.setItemNumber(1, columnIn, Double.valueOf(newValueIn)); } return newValueIn; import com.agileassetsinc.core.TRDIEException; if(dslIn.getItemNumber(1, "MATERIAL_TRANSACTION_TYPE_ID ") != 5) { if(Double.valueOf(newValueIn) < 0) { throw new TRDIEException(237); } else { dslIn.setItemNumber(1, columnIn, Double.valueOf(newValueIn)); } } else { dslIn.setItemNumber(1, columnIn, Double.valueOf(newValueIn)); } return newValueIn; import com.agileassetsinc.core.*; String[] Columns = ["OWNER_ID", "EFF_YEAR", "BUDGET_CAT_ID"]; HashMap<String, Integer> Values = new HashMap<String, Integer>(); for(String S: Columns) { Values.put(S, dslIn.getItemInteger(1, S)); } Values.put(columnIn, Integer.parseInt(newValueIn)); </pre>
23	UNIT_COST_CHANGE	
65	Update QG Cost	

ID	Groovy Script Name	Groovy Script Text
		<pre> Integer[] args = new Integer[Columns.length]; boolean hasNulls = false; int i = 0; for(String S: Columns) { args[i] = Values.get(S); if(args[i] == null) { hasNulls = true; } i++; } if(!hasNulls) { String sql = "select sum(total_cost) from master_matrix_totals_vw b where b.owner_id = ? and b.eff_fiscal_year = ? " + " and b.budget_cat_id = ?"; DataStore ds = dl.createDataStore(sql); if(ds.retrieve(args[0], args[1], args[2]) > 0) { Double cost = ds.getItemNumber(1,1); if(cost != null) { dsIn.setItemNumber(1,"QG_COST", cost); } else { dsIn.setItemNull(1,"QG_COST"), } } } </pre>
98	WR_TYPE_CHECK	<pre> import com.agileassetsinc.core.*; Double REQUEST = Double.valueOf(new ValueIn); if(dsIn.getItemNumber(1,"WR_APPROV ED") == 1) { throw new TRDException("Cannot change Approved record"); } if(dsIn.getItemNumber(1,"ID_WR_CATE GORY_ID") != null && REQUEST != 2) { dsIn.setItemNull(1,"ID_WR_CATEGORY _ID"); return "basic_window.ShowAlert('NOTE: Since the Work Request Type selected is not *Emergency*, the Emergency Category value is being set to Null.');" } </pre>

ID	Groovy Script Name	Groovy Script Text
55	Set District to Parent	<pre> import com.agileassetsinc.core.*; String sql = "select owner_id, level_id from setup_owner connect by prior parent_id = owner_id start with owner_id = ?"; DataStore ds = dl.createDataStore(sql); int rowCount = ds.retrieve(Integer.parseInt(newValueIn)); for(int row = 1; row <= rowCount; row++) { if ((3).equals(ds.getItemInteger(row, "LEVEL_ID"))) { Integer ParentId = ds.getItemInteger(row,"OWNER_ID"); dsIn.setItemNumber(1,"DISTRICT_ID", ParentId); } } String[] Columns = ["OWNER_ID", "EFF_YEAR", "BUDGET_CAT_ID"]; HashMap<String, Integer> Values = new HashMap<String, Integer>(); for(String S: Columns) { Values.put(S, dsIn.getItemInteger(1, S)); } Values.put(columnIn, Integer.parseInt(newValueIn)); Integer[] args = new Integer[Columns.length]; boolean hasNulls = false; int i = 0; for(String S: Columns) { args[i] = Values.get(S); if(args[i] == null) { hasNulls = true; } i++; } if(!hasNulls) { sql = "select sum(total_cost) from master_matrix_totals_vw b where b.owner_id = ? and b.eff_fiscal_year = ? " + " and b.budget_cat_id = ?"; ds = dl.createDataStore(sql); if(ds.retrieve(args[0], args[1], args[2]) > 0) { Double cost = </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> if(cost != null) { dsIn.setItemNumber(1,"QG_COST", cost); } else { dsIn.setItemNull(1, "QG_COST"); } } </pre>
54	Budget Plan Window	<pre> import com.agileassetsinc.core.TRDIEException; if(Double.valueOf(newValueIn) != null) { throw new TRDIEException(237); } else { dsIn.setItemNumber(1, "CI_100", Double.valueOf(newValueIn)*20); } ; select OWNER_ID from Setup_owner where owner_id = (select parent_id from setup_owner where owner_id = newValueIn and level_id = 4) union aLL select newValueIn from SETUP_OWNER where 1 = (case when (select 1 from setup_owner where owner_id = newValueIn and level_id = 3) = 1 then 1 else 2 end) and owner_id = newValueIn </pre>
45	Fill CALENDAR_ID by CREW_SUB_ID	<pre> import com.agileassetsinc.core.*; if(newValueIn!=null) { DataStore dsCalendar = dl.createRetrieveDataStore("select CALENDAR_ID from LABOR_CREWS where CALENDAR_ID is not NULL and CREW_ID=?",null,newValueIn); if(dsCalendar.rowCount()==1) { dsIn.setItemNumber(1,"CALENDAR_ID", dsCalendar.getItemInteger(1,1)); } } </pre>
89	onEditQAReqStatus	<pre> import com.agileassetsinc.core.* /* when editing the qa status: If original value: Was Null or Reviewed then prevent editing at all, only QA request should change value of status Was Pending Then new value may only be </pre>

ID	Groovy Script Name	Groovy Script Text
		Reviewed or pending; After passing edit checks if newValue is Reviewed then Set QA Request to 0 Reviewdate to sysdate Review User to Current User (Required Comments enforced on save) Pending then do nothing */ if (dsln.getItemObjectOriginal(1,columnIn) ==null dsln.getItemObjectOriginal(1,columnIn)= =3) { throw new TRDException(1243); //Change the status of this record by requesting a QA review first. } else if (newValueIn==null) { throw new TRDException(1244); // You cannot set the status of a previously reviewed record to null. } if (newValueIn=="3") { dsln.setItemNumber(1,"ID_QA_REVIEW ",0); //Clear QA Request Checkbox.. dsln.setItemDate(1,"ID_QA_REVIEW_D ATE", new DateTime()); //Set to today dsln.setItemString(1,"ID_QA_REVIEWE R", dl.getSession().getUserId()); //Set to current user } import com.agileassetsinc.core.*; /*When Changing Value of from milepost lookup the segment code Code works on edit of ID_MILEPOST_FROM and ID_MILEPOST_TO columns*/ Integer locRefId = dl.getSession().getLocRefId(); /* Only operate is not in Basic Location Reference*/ int route_from; String sql_str; boolean from_edit; DataStore ds; if(columnIn=="ID_MILEPOST_FROM") { from_edit=true; } else { from_edit=false; } route_from = dsln.getItemInteger(1,"
80	Set Segment on Edit Milepost	

ID	Groovy Script Name	Groovy Script Text
		<pre> ID_ROUTE_FROM"); if (!LRS.getInstance().getLocRef(locRefId) .isBasic()) { if(newValueIn != "" && route_from != null) { if (from_edit) { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM<=" + newValueIn + " and a.ID_MILEPOST_TO> "+ newValueIn + " and source_table='LOC_REF_42'"; } else { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + route_from + " and a.ID_MILEPOST_FROM< "+ newValueIn + " and a.ID_MILEPOST_TO>=" + newValueIn + " and source_table='LOC_REF_42'"; } ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0) { throw new TRDException("The milepost "+newValueIn+ " was not found on route "+dsIn.getColumn("ID_ROUTE_FROM") .getLookup().lookupLabelById(dl, route_from)); } if(ds.getItemInteger(1,"CNT")>1) { if(from_edit) { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); } else { dsIn.setItemString(1,"ID_SEGMENT_TO ",""); } dl.getSession().setWarningString("The </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> lookupLabelById(dl, route_from)+ ", you must specify the desired segment code."); } else { if(from_edit) { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<=" + newValueln + " and a.ID_MILEPOST_TO> "+ newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_FR OM",ds.getItemString(1,"ID_SEGMENT _FROM")); } else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ route_from + " and a.ID_MILEPOST_FROM<" + newValueln + " and a.ID_MILEPOST_TO>=" + newValueln + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_TO ",ds.getItemString(1,"ID_SEGMENT_FR OM")); } } } </pre>
81	ON_EDIT_ID_ROUTE_FROM	<pre> } import com.agileassetsinc.core.*; /*When Changing Value of from route check to see if milepost is valid, if not make milepost from and to null, otherwise leave alone and set segment properly for from and to) Code works on edit of ID_ROUTE_FROM columns*/ Integer locRefId = dl.getSession().getLocRefId(); /* Only operate is not in Basic </pre>

ID	Groovy Script Name	Groovy Script Text
		<p>Location Reference*/</p> <pre> String sql_str; boolean from_edit; DataStore ds; Double mp_from; Double mp_to; String seg_from; String seg_to; mp_from = dsIn.getItemInteger(1,"ID_MILEPOST_F ROM"); mp_to = dsIn.getItemInteger(1,"ID_MILEPOST_T O"); seg_from=dsIn.getItemInteger(1,"ID_SE GMENT_FROM"); seg_to=dsIn.getItemInteger(1,"ID_SEG MENT_TO"); if (!LRS.getInstance().getLocRef(locRefId) .isBasic()) { if(newValueIn != "" && mp_from != "") { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + newValueIn + " and a.ID_MILEPOST_FROM<= " + mp_from + " and a.ID_MILEPOST_TO> " + mp_from + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0 ds.get ItemInteger(1,"CNT")>1) { dsIn.setItemString(1,"ID_SEGMENT_FR OM",""); dsIn.setItemString(1,"ID_MILEPOST_FR OM",""); } else {sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ newValueIn + " and a.ID_MILEPOST_FROM<= " + mp_from + " and a.ID_MILEPOST_TO> " + mp_from + " and source_table='LOC_REF_42'"; </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> dsIn.setItemString(1,"ID_SEGMENT_FROM",ds.getItemString(1,"ID_SEGMENT_FROM")); } } if(newValueIn != "" && mp_to != "") { sql_str="select count(distinct a.ID_SEGMENT_FROM) as CNT from setup_loc_ident a " + " where a.ID_ROUTE_FROM=" + newValueIn + " and a.ID_MILEPOST_FROM<=" + mp_to + " and a.ID_MILEPOST_TO> "+ mp_to + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); if(ds.getItemInteger(1,"CNT")==0 ds.getItemInteger(1,"CNT")>1) { dsIn.setItemString(1,"ID_SEGMENT_TO", ""); dsIn.setItemString(1,"ID_MILEPOST_TO", ""); } else { sql_str= " select a.ID_SEGMENT_FROM from setup_loc_ident a where a.ID_ROUTE_FROM="+ newValueIn + " and a.ID_MILEPOST_FROM<=" + mp_to + " and a.ID_MILEPOST_TO> "+ mp_to + " and source_table='LOC_REF_42'"; ds = dl.createRetrieveDataStore(sql_str); dsIn.setItemString(1,"ID_SEGMENT_TO",ds.getItemString(1,"ID_SEGMENT_FROM")); } } } import com.agileassetsinc.core.*; Double num = TRDIUtils.strToDouble(newValueIn); Double rem; rem = num % 0.5; // Double sql; // sql = Math.ceil(num*2)/2; // dsIn.setItemNumber(1, columnIn,</pre>
102	nearest_half_check	

ID	Groovy Script Name	Groovy Script Text
		sql);
		if (rem!=0) { throw new TRDIException ("Please enter time in half hour increments only."); }
160	Work Date Check on Acc	import com.agileassetsinc.core.* if(dsIn.getItemNumber(1,"DATE_WORK") == null) { throw new TRDIException("Please enter 'Work Date' before accessing Journal Entry"); }
324	Send Loc_ident to iVision	/*import com.agileassetsinc.core.*; import com.agileassetsinc.core.TRDIUtils; import com.vividsolutions.jts.geom.Geometry; import com.vividsolutions.jts.geom.Coordinate; import com.agileassetsinc.core.gisnew.GISUtils ; Integer networkThemeld = LRS.getInstance().getLineNetwork(dl.get Session().getNetworkId()).getThemeld(); String iVisionLink="http://ivision.ucprc.ucdavis. edu/iVision/#!/Home?year=2010&"; if(!dsIn.isValidColumn("CA_ROUTE_FR OM")) return ""; if(networkThemeld == null) return ""; if(!dsIn.hasLocation()) return ""; Geometry geometry = dsIn.getLocation(1).toBasic(dl).toGeome try(dl); if(geometry == null) return ""; Coordinate coordinate = GISUtils.toWGS84(geometry.getCoordin ates()[0], GISUtils.createGisLayer(dl, networkThemeld).getCoordinateSystem()); // return "http://169.237.179.14/iVision/#!/Home?y ear=2010&gpslat="+coordinate.y+"&gpsl ong="+coordinate.x; //Add Coords iVisionLink = iVisionLink + "gpslat="

ID	Groovy Script Name	Groovy Script Text
		<pre> +coordinate.y+"&gpslong="+coordinate. x; //Add District iVisionLink=iVisionLink+"&district_="+TR DIUtils.leftPad(dsIn.getItemString(1,"CA _DISTRICT_ID"),2,'0' as char); //Add County iVisionLink=iVisionLink+"&county_="+ds In.getColumn("CA_COUNTY_FROM").g etLookup().lookupLabelById(dl, dsIn.getItemInteger(1, "CA_COUNTY_FROM")); //Add Route and suffix iVisionLink=iVisionLink+"&route_="+TR DIUtils.leftPad(dsIn.getItemString(1,"CA _ROUTE_FROM"),3,'0' as char)+(dsIn.getItemString(1,"CA_RTE_S UFFIX_FROM")==null?"":dsIn.getItemSt ring(1,"CA_RTE_SUFFIX_FROM")); //Add Direction iVisionLink=iVisionLink+"&direction_="+ dsIn.getItemString(1,"CA_DIRECTION_ F_NAME").substring(0,1)+"B"; //Add Lane if(dsIn.getItemString(1,"LANE_ID")==="0") { iVisionLink=iVisionLink+"&lane_=1"; } else { iVisionLink=iVisionLink+"&lane_="+dsIn. getItemString(1,"LANE_ID"); } //Add Guest user iVisionLink=iVisionLink+"&guest"; return iVisionLink; */ import com.agileassetsinc.core.*; import com.agileassetsinc.core.TRDIUtils; import com.vividsolutions.jts.geom.Geometry; import com.vividsolutions.jts.geom.Coordinate; import com.agileassetsinc.core.gisnew.GISUtils ; Integer networkThemeld = LRS.getInstance().getLineNetwork(dl.get Session().getNetworkId()).getThemeld(); String iVisionLink="http://ivision.ucprc.ucdavis. edu/iVision/#!/Home?year=2010&"; if(networkThemeld == null){ throw new TRDIException("Network </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> theme not established. Please contact your system administrator."); } if(!dsIn.hasLocation()) { throw new TRDIException("No location defined. Please contact your system administrator."); } Geometry geometry = dsIn.getLocation(1).toBasic(dl).toGeome try(dl); if(geometry == null) { throw new TRDIException("Location does not have a geometry. Please contact your system administrator."); } Coordinate coordinate = GISUtils.toWGS84(geometry.getCoordin ates()[0], GISUtils.createGisLayer(dl, networkThemeld).getCoordinateSystem()); // return "http://169.237.179.14/iVision/#/Home?y ear=2010&gpslat="+coordinate.y+"&gpsl ong="+coordinate.x; //Add Coords iVisionLink = iVisionLink + "gpslat="+coordinate.y+"&gpslong="+co ordinate.x; //Add District if(dsIn.isValidColumn("CA_DISTRICT_I D")) { iVisionLink=iVisionLink+"&district_="+TR DIUtils.leftPad(dsIn.getItemString(1,"CA _DISTRICT_ID"),2,'0' as char); } //Add County if(dsIn.isValidColumn("CA_COUNTY_F ROM")) { iVisionLink=iVisionLink+"&county_="+ds In.getColumn("CA_COUNTY_FROM").g etLookup().lookupLabelById(dl, dsIn.getItemInteger(1, "CA_COUNTY_FROM")); } //Add Route and suffix if(dsIn.isValidColumn("CA_ROUTE_FR OM")) { iVisionLink=iVisionLink+"&route_="+TR DIUtils.leftPad(dsIn.getItemString(1,"CA _ROUTE_FROM"),3,'0' as char)+(dsIn.getItemString(1,"CA_RTE_S UFFIX_FROM")==null?"":dsIn.getItemSt </pre>

ID	Groovy Script Name	Groovy Script Text
		CA_RTE_SUFFIX_FROM")); } //Add Direction if(dsIn.isValidColumn("CA_DIRECTION_ F_NAME")) { iVisionLink=iVisionLink+"&direction_="+ dsIn.getItemString(1,"CA_DIRECTION_ F_NAME").substring(0,1)+"B"; } //Add Lane if(dsIn.getItemString(1,"LANE_ID")==0") { iVisionLink=iVisionLink+"&lane_=1"; } else { iVisionLink=iVisionLink+"&lane_="+dsIn. getItemString(1,"LANE_ID"); }
		//Add Guest user iVisionLink=iVisionLink+"&guest"; return iVisionLink;
43	40 Threshold	Double ret; ret=40; return ret;
314	IRI RSL Threshold	Double ret; ret=250; return ret;
227	2.0 INT 1.0 Else	import com.agileassetsinc.core.*; Double ret; if (ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==1 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==2 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==6 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==11 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==14 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==16 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==12) { ret =2.0; } else { ret=1.0; } return ret;
-17	General: 40 for Int, 30 for P, else 20	import com.agileassetsinc.core.*; Double ret; Integer val = dsIn.getItemInteger(rowIn,"NC_SYSTE M_CODE_ID");

ID	Groovy Script Name	Groovy Script Text
		<pre> { ret = new Double(40); } else if (val != null && val.intValue() <= 3) { ret = new Double(30); } else { ret = new Double(20); } return ret; import com.agileassetsinc.core.*; Double ret; if (ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==1 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==2 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==6 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==11 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==14 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==16 ds_in.getItemInteger(row_in,"ID_FUNC_ CLASS_ID")==12) { ret =2.5; } else { ret=2.0; } return ret; import com.agileassetsinc.core.*; //Power Value formula Double ret = var3 + var1 * Math.pow(varln,var2); return Math.min(100,Math.max(0, ret)); </pre>
133	2.5 for Ints/Art 2.0 else	
16	Power Value	
257	257	null
226	226	null
177	Weelpath Cracking Value	<pre> import com.agileassetsinc.core.*; //Cal Flexible Total Cracking Index //100 - 100*(1-EXP(-((Age/a)^b))) Double ret=Math.pow((varln/var1),var2);/(Age/a)^b ret=100 * Math.exp(-ret);/1-EXP(- ((Age/a)^b)) return Math.min(100,Math.max(0, ret)); </pre>

ID	Groovy Script Name	Groovy Script Text
176	Weelpath Cracking Root	<pre>// a*(Ln(100)-Ln(Value))^(1/b) Double ret=Math.log(100) - Math.log(var1n) ,/(-ln(Value)+ln(100)) ret = var1 * Math.pow(ret,1/var2); return Math.min(100,Math.max(0, ret));</pre>
17	Sigmoidal Root	<pre>import com.agileassetsinc.core.*; //Sigmoidal Root formula if(var1 == 0 var3 == 0){ throw new TRDException("Divide by zero error in Sigmoidal Root formula"); } if(((var1n - var4) / var1) >= 1){ return new Double(50.0); } else if(((var1n - var4) / var1) <= 0){ return new Double(0.0); } else{ Double ret = var2 / Math.pow(- Math.log((var1n - var4) / var1),(1/var3)); return Math.min(100,Math.max(0, ret)); }</pre>
18	Sigmoidal Value	<pre>import com.agileassetsinc.core.*; //Sigmoidal Value formula Double ret = var4 + var1 * Math.exp(- Math.pow((var2/var1n),var3)); return Math.min(100,Math.max(0, ret));</pre>
19	Piecewise Linear Root	<pre>import com.agileassetsinc.core.*; //Piecewise Linear Root formula Double ret; if (var1n >= var2){ ret = var1; } else if (var1n >= var4 && var1n < var2){ if(var4 - var2 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var1 + (var1n - var2)*(var3 - var1)/(var4 - var2); } else if (var1n >= var6 && var1n < var4){ if(var6 - var4 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var3 + (var1n - var4)*(var5 -</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> var3)/(var6 - var4); } else if (varIn >= var8 && varIn < var6){ if(var8 - var6 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var5 + (varIn - var6)*(var7 - var5)/(var8 - var6); } else if (varIn >= var10 && varIn < var8){ if(var10 - var8 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var7 + (varIn - var8)*(var9 - var7)/(var10 - var8); } else if (varIn >= var12 && varIn < var10){ if(var12 - var10 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var9 + (varIn - var10)*(var11 - var9)/(var12 - var10); } else if (varIn >= var14 && varIn < var12){ if(var14 - var12 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Root formula"); } ret = var11 + (varIn - var12)*(var13 - var11)/(var14 - var12); } else{ ret = var13; } return Math.min(100,Math.max(0, ret)); </pre>
7	Linear Root	<pre> import com.agileassetsinc.core.*; //Linear Root formula if(var2 == 0){ throw new TRDException("Divide by zero error in Linear Root formula"); } Double ret = (varIn-var1) / var2; return Math.min(100,Math.max(0, ret)); </pre>

ID	Groovy Script Name	Groovy Script Text
8	Linear Value	<pre>import com.agileassetsinc.core.*; //Linear Value formula Double ret = var1+var2*varln; return Math.min(100,Math.max(0, ret));</pre>
9	Exponential Root	<pre>import com.agileassetsinc.core.*; //Exponential Root formula if(var1 == 0 var2 == 0){ throw new TRDIEException("Divide by zero error in Exponential Root formula."); } if((varln-var3)/var1 > 0){ Double ret = Math.log((varln-var3) / var1) / var2; return Math.min(100,Math.max(0, ret)); } else{ return new Double(0.0); }</pre>
10	Exponential Value	<pre>import com.agileassetsinc.core.*; //Exponential Value formula Double ret = var3+var1*Math.exp(var2*varln); return Math.min(100,Math.max(0, ret));</pre>
11	Inverse Exponential Root	<pre>import com.agileassetsinc.core.*; //Inverse Exponential Root formula if(var1 == 0 var2 == 0){ throw new TRDIEException("Divide by zero error in Inverse Exponential Root formula"); } if((varln-var3)/var1 > 0){ Double ret = Math.log((varln- var3)/var1)/var2; return Math.min(100,Math.max(0, ret)); } else{ return new Double(0.0); }</pre>
12	Inverse Exponential Value	<pre>import com.agileassetsinc.core.*; //Inverse Exponential Value formula Double ret = var3+var1*Math.exp(var2*varln); return Math.min(100,Math.max(0, ret));</pre>
13	Hyperbolic Root	<pre>import com.agileassetsinc.core.*; //Hyperbolic Root formula if(var2 == 0){ throw new TRDIEException("Divide</pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> by zero error in Hyperbolic Root formula"); } Double ret = (1/(varIn-var3) - var1) / var2; return Math.min(100,Math.max(0, ret)); </pre>
14	Hyperbolic Value	<pre> import com.agileassetsinc.core.*; //Hyperbolic Value formula if(var1+var2 == 0){ throw new TRDException("Divide by zero error in Hyperbolic Value formula"); } Double ret = var3+1/(var1+var2*varIn); return Math.min(100,Math.max(0, ret)); </pre>
15	Power Root	<pre> import com.agileassetsinc.core.*; //Power Root formula if(var1 == 0 var2 == 0){ throw new TRDException("Divide by zero error in Power Root formula"); } Double ret = 0.0; if(varIn > var3){ return ret; } ret = Math.pow(((varIn - var3) / var1),(1/var2)); return Math.min(100,Math.max(0, ret)); </pre>
20	Piecewise Linear Value	<pre> import com.agileassetsinc.core.*; //Piecewise Linear Value formula Double ret; if (varIn <= var1){ ret = var2; } else if (varIn > var1 && varIn <= var3){ if(var3 - var1 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var2 + (varIn - var1)*(var4 - var2)/(var3 - var1); } else if (varIn > var3 && varIn <= var5){ if(var5 - var3 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var4 + (varIn - var3)*(var6 - </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre> var4)/(var5 - var3); } else if (varLn > var5 && varLn <= var7){ if(var7 - var5 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var6 + (varLn - var5)*(var8 - var6)/(var7 - var5); } else if (varLn > var7 && varLn <= var9){ if(var9 - var7 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var8 + (varLn - var7)*(var10 - var8)/(var9 - var7); } else if (varLn > var9 && varLn <= var11){ if(var11 - var9 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var10 + (varLn - var9)*(var12 - var10)/(var11 - var9); } else if (varLn > var11 && varLn <= var13){ if(var13 - var11 == 0){ throw new TRDException("Divide by zero error in Piecewise Linear Value formula"); } ret = var12 + (varLn - var11)*(var14 - var12)/(var13 - var11); } else{ ret = var14; } return Math.min(100,Math.max(0, ret)); </pre>
21	Universal Power Root	<pre> //Cal Flexible Total Cracking Index // a*((-Ln(1-(c/100))*((Ln(100)- Ln(Value))/Ln(2))))^(1/b) Double ret= var1 *Math.pow ((- Math.log(1 +((Math.log(varLn) - Math.log(100))/Math.log(2))*(var3/100))), 1/var2); return Math.min(100,Math.max(0, ret)); </pre>
22	Universal Power Value	<pre> import com.agileassetsinc.core.*; //Cal Flexible Total Cracking Index </pre>

ID	Groovy Script Name	Groovy Script Text
		<pre>//100*2^-((1-EXP(-(Age/a)^b))*100/7.5)) Double ret=Math.pow((varln/var1),var2);//(Age/a)^b ret=1 - Math.exp(-ret);//1-EXP(-(Age/a)^b) ret =-100/ var3 *ret ;//-(1-EXP(-(Age/a)^b))*100/7.5)</pre>
186	System Job Notification	<pre>ret = 100 * Math.pow(2,ret); return Math.min(100,Math.max(0, ret)); import com.agileassetsinc.core.*; import com.agileassetsinc.core.cache.SystemAdmCache; import java.util.Set; int j = dsln.rowCount(); for(int i=1; i<=j; i++) { int systemJobId = dsln.getItemInteger(i, "SYSTEM_JOB_ID"); String systemJobName = dsln.getColumn("SYSTEM_JOB_ID").getLookup().lookupLabelById(dl, systemJobId); Set<String> userIds = SystemNotification.groupToUsers(dl, notificationTypeId, systemJobId); String title = "System Job"; if(dsln.isRowNew(i)) { /*system job started*/ /* SystemNotification.send(dl, notificationTypeId, userIds, title, "System Job " + systemJobName + " started at " + TRDIUtils.dateToStr (dsln.getItemDate(i, "SYSTEM_JOB_LOG_START_DATE"), SystemSetup.getInstance().getTimeFor mat()); */ } else { /*system job ended*/ SystemNotification.send(dl, notificationTypeId, userIds, title, "System Job " + systemJobName + " finished at " + TRDIUtils.dateToStr (dsln.getItemDate(i, "SYSTEM_JOB_LOG_END_DATE"), SystemSetup.getInstance().getTimeFor mat()) + (dsln.isItemNull(i, "ERROR_TEXT")) ? "successfully" : " unsuccessfully with the following error:\n" + dsln.getItemString(i, "ERROR_TEXT"));</pre>
40	Treatment Script	<pre>import com.agileassetsinc.core.*; throw new TRDIException("treatment groovy script not made yet");</pre>