



EXPERTISE

SAFETY

SOLUTIONS

Rave Web Services (RWS) Approach for Rollover trials: Reducing the Burdens

Tom De Coster – Team Manager eSolutions, SGS Health Science

PHUSE US Connect 2021 | 14-18 June 2021

WHEN YOU NEED TO BE SURE

SGS

DISCLAIMER AND DISCLOSURES

- *The views and opinions expressed in this presentation are those of the author(s) and do not necessarily reflect the opinions of PHUSE members, respective companies or organizations.*



RWS

- Rave Web Services
- Demo
- Custom Functions Code
- Conclusion

- Integrates Medidata Rave EDC with other systems
- Using CDISC ODM standard clinical data or metadata
- Synchronously and with immediate confirmation

- Subjects can roll-over from parent trial to follow-up trial
- Only subjects from parent trial(s)
- Last visit parent trial = first visit follow-up trial
 - Same data needs to be entered twice in EDC

- Site users only need to complete a trigger question in the parent trial
- Auto-creation of the subject in follow-up trial
- Applicable subject data copied to follow-up trial

- Client to decide if copied data needs to remain editable or not
 - To reduce reconciliation, copied data should not be editable in the target trial
 - Data changes would occur in parent trial and a copy action should be subsequently triggered
 - Alternatively, if copied data remains editable in follow-up trial, data changes need to be made twice

- Integrate Rave EDC parent trial with follow-up trial using RWS
 - Custom Functions generate the necessary ODM XMLs from the parent trial
 - Custom Function sends the XMLs via RWS to the follow-up trial
 - RWS instantly returns confirmation messages

```
string sbjName = sbj.Name.ToString();
string siteNumber = sbj.StudySite.Site.Number;
string studyname = "DemoFollowUp";
string studyoid = studyname;
string env = "(" + sbj.StudySite.Study.Environment.ToString() + ")";
string odmSUB = "", odmDM = "", odmDOV = "", odmMH = "", odmMGH = "", odmAE = "", odmPROCED = "",
odmCM = "", odmMGADL = "", odmQMG = "", odmPHQ9 = "", odmPE = "",
odmVS2 = "", odmECG = "", odmLABCBL = "", odmLABCUR = "", odmPD = "", odmADA = "", odmPREG = "";
if (sbj.StudySite.Study.TestStudy == true) studyoid += env;
//Createsbj
odmSUB = (string) CustomFunction.PerformCustomFunction("_GetCreateSubject", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmSUB != "") blok = CanPostRequest(odmSUB, dp);
if (blok)
{
    env DM
```

```
string odmTemplate = @"<ODM xmlns=""http://www.cdisc.org/ns/odm/v1.3"" ODMVersion=""1.3"" FileType=""Transactional"" FileOID=""SUBJECT""
CreationDateTime=""2018-01-01T00:00:00"">
<ClinicalData StudyOID = ""
{
    0
}
"" MetaDataVersionOID = ""122"">
<SubjectData SubjectKey = ""
{
    1
}
"" TransactionType = ""Insert"">
<SiteRef LocationOID = ""
{
    2
}
""/>
<StudyEventData StudyEventOID = ""SUBJECT"" TransactionType = ""Update"">
<FormData FormOID = ""SUBJECT"" TransactionType = ""Update"">
<ItemGroupData ItemGroupOID = ""SUBJECT"">
<ItemData ItemOID = ""SUBJID"" Value = ""
{
    1
}
"" TransactionType = ""Update""/>
<ItemData ItemOID = ""SITEID"" Value = ""
{
    3
}
"" TransactionType = ""Update""/>
</ItemGroupData>
</FormData>
</StudyEventData>
</SubjectData>
</ClinicalData>
</ODM>";
String odm = String.Format(odmTemplate, studyoid, subjectName, siteNumber, SiteID);
// replace parameters
return odm;
```

```
private bool CanPostRequest(string odmxml, DataPoint dpAct)
{
    bool Success = false;
    const int TIME = 60000;
    string RWSResponse = string.Empty;
    const string url = "https://sgs-sandbox.mdsol.com/RaveWebServices/WebService.aspx?PostODMClinicalData", user = "xxx", pass = "xxx";
    IConnector connector = new Connector();
    IHttpRequestWithBody request = connector.CreateHttpPostRequest(url);
    if (request != null)
    {
        request.AddHeader("Content-Type", "text/xml");
        request.SetBasicAuthenticationCredentials(user, pass);
        request.Timeout = TIME;
        request.AppendToBody(odmxml);
        try
        {
            IHttpResponse response = request.GetResponse();
            if (response != null)
            {
                RWSResponse = response.Code + ":" + response.ReadBody();
                if (RWSResponse.StartsWith("OK")) Success = true;
            }
        }
        catch (HttpRequestException e)
        {
            RWSResponse = e.Response.ReadBody();
        }
        DataPoint dpResponse = null;
        if (dpAct != null) dpResponse = dpAct.Record.DataPoints.FindByFieldOID("ROLLRWS");
        if (dpResponse != null)
        {
            if (RWSResponse != null && RWSResponse != string.Empty)
            {
                RWSResponse = TrimmedResponse(RWSResponse);
                //trim response
                EnterData(dpResponse, RWSResponse);
                //enter response
            }
        }
    }
    return Success;
}
```

```
//copy DM
odmDM = (string) CustomFunction.PerformCustomFunction("_GetDMData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmDM != "") bLOK = CanPostRequest(odmDM, dp);
//copy DOV from EOS
odmDOV = (string) CustomFunction.PerformCustomFunction("_GetDOVData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmDOV != "") bLOK = CanPostRequest(odmDOV, dp);
//----from SCREEN----
//copy MH
odmMH = (string) CustomFunction.PerformCustomFunction("_GetMHData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmMH != "") bLOK = CanPostRequest(odmMH, dp);
//copy MGH
odmMGH = (string) CustomFunction.PerformCustomFunction("_GetMGHData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmMGH != "") bLOK = CanPostRequest(odmMGH, dp);
//---from SCRDS---
//copy AE
odmAE = (string) CustomFunction.PerformCustomFunction("_GetAEDData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmAE != "") bLOK = CanPostRequest(odmAE, dp);
//copy PROCED
odmPROCED = (string) CustomFunction.PerformCustomFunction("_GetPROCEDData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmPROCED != "") bLOK = CanPostRequest(odmPROCED, dp);
//copy CM
odmCM = (string) CustomFunction.PerformCustomFunction("_GetCMDData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmCM != "") bLOK = CanPostRequest(odmCM, dp);
//----from EOS----
//copy MGADL
odmMGADL = (string) CustomFunction.PerformCustomFunction("_GetMGADLData", sbj.CRFVersion.ID, new object[] { studyoid, sbjName, siteNumber, sbj } );
if (odmMGADL != "") bLOK = CanPostRequest(odmMGADL, dp);
```

```
String odm = "";
//get the datapoints to copy
DataPoint dpBRTH = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID, new object[] { sub, "SCREEN", "", "DM", "BRTH" } );
if (dpBRTH != null && dpBRTH.Active)
{
    DataPoint dpSEX = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "SEX" } );
    DataPoint dpCHILDPOT = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "CHILDPOT" } );
    DataPoint dpCHILDPOTN = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "CHILDPOTN" } );
    DataPoint dpMENSDT = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "MENSDT" } );
    DataPoint dpRACEAIAN = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACEAIAN" } );
    DataPoint dpRACEA = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACEA" } );
    DataPoint dpRACEBA = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACEBA" } );
    DataPoint dpRACENHOP = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACENHOP" } );
    DataPoint dpRACEW = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACEW" } );
    DataPoint dpRACEO = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
    new object[] { sub, "SCREEN", "", "DM", "RACEO" } );
    DataPoint dpRACEOSP = (DataPoint) CustomFunction.PerformCustomFunction("_GetDataPoint", sub.CRFVersion.ID,
```

```
//get the data values
string BRTH = dpBRTH.Data;
string SEX = dpSEX.Data;
string CHILD POT = dpCHILDPOT.Data;
string CHILDPOTN = dpCHILDPOTN.Data;
string MENS DT = dpMENS DT.Data;
string RACEAIAN = dpRACEAIAN.Data;
string RACEA = dpRACEA.Data;
string RACEBA = dpRACEBA.Data;
string RACENHOP = dpRACENHOP.Data;
string RACEW = dpRACEW.Data;
string RACEO = dpRACEO.Data;
string RACEOSP = dpRACEOSP.Data;
string RACENR = dpRACENR.Data;
string ETHNIC = dpETHNIC.Data;
```

```
//ODM XML
string odmTemplate = @"<ODM xmlns=""http://www.cdisc.org/ns/odm/v1.3"" ODMVersion=""1.3"" FileType=""Transactional"" FileOID=""DM""
CreationDateTime=""2018-01-01T00:00:00"">
<ClinicalData StudyOID = ""
{
  0
}
"" MetaDataVersionOID = ""122"">
  <SubjectData SubjectKey = ""
  {
    1
  }
  "" TransactionType = ""Update"">
    <SiteRef LocationOID = ""
    {
      2
    }
    ""/>
    <StudyEventData StudyEventOID = ""SEB"">
      <FormData FormOID = ""DM"">
        <ItemGroupData ItemGroupOID = ""DM"">
          <ItemData ItemOID = ""BRTH"" Value = ""
          {
            3
          }
          ""/>
          <ItemData ItemOID = ""SEX"" Value = ""
          {
            4
          }
          ""/>
        </ItemGroupData>
      </FormData>
    </StudyEventData>
  </SubjectData>
</ClinicalData>
</ODM>";
odm = String.Format(odmTemplate, studyoid, subjectName, siteNumber, BRTH, SEX, CHILDPOT, CHILDPOTN, MENSMT, RACEAIAN, RACEA, RACEBA,
RACEOSP, RACENR, ETHNIC);
// replace parameters
```

```
if (user != null)
{
    strUser = user.FullName;
    if (strUser.Length > 200) strUser = strUser.Substring(0, 199);
    DataPoint dpUser = dp.Record.DataPoints.FindByFieldOID("ROLLINIT");
    EnterData(dpUser, strUser);
}
string NOW = DateTime.Now.Date.ToString("dd MMM yy");
DataPoint dpDate = dp.Record.DataPoints.FindByFieldOID("ROLLDT");
if (dpDate != null) EnterData(dpDate, NOW);
}
}
catch (Exception ex)
```



- Reduces data entry burden for site staff
- Eliminates data entry mistakes
- Allows to control subject generation in follow-up trial



- Reduces reconciliation effort
- Automated process: triggered by the site, quick
- Can be built upfront in parent trial and activated using Publish Checks



- Define upfront which data to be copied (detailed)
- Ensure similar study design
- Adverse Events, Concomitant Medications, Medical History
→ "ongoing"



- Allow multiple “copy moments”
 - + Eliminate the need to make the same edits twice after roll-over
 - ⚠ Increased complexity

 BEL9991003  Screening to end of trial

 Roll-Over

Requires Verification

Will the patient roll over to the Follow-Up trial? Yes

If the patient will roll over to the Follow-Up trial, please check this box to transfer data ☒ 

Date data transferred (dd MMM yyyy) 29 Oct 2019 

Trigger action performed by Liesbeth Renneboog 

RWS return value

SUBJECT: OK;DM: OK;DOV: OK;MH: OK; OK;AE: OK;CM: OK; ADL: OK;PHQ9: OK;PE: OK;VS2: OK;ECG: OK;LABCBL: OK;LABCUR: OK;PD: OK;ADA: OK 

- Capture roll-over trigger information in parent trial
→ user, date and time
- Capture RWS response
→ one response per ODM XML sent

EXPERTISE

- Copied data can trigger edit checks
→ e.g. email alert to sponsor users, edit checks

The screenshot displays the SGS Hospital 1 EDC system interface. The top navigation bar includes tabs for STUDIES (Fsgs-sandbox, DemoFollowUp), ACTIONS (EDC, Production), SITES (BEL9991 - SGS Hospital 1), and SUBJECTS (BEL9991003). The sidebar on the left lists various data collection points: Subject Status, Screening, Informed Consent Recording, Eligibility, Medical History (selected), Disease History, Activities of Daily Living - ADL, Disease Q 1, Patient Health Questionnaire Item 9 (Suicidality assessment), Physical Examination, Vital Signs 2, ECG, Central Laboratory - Blood, Central Laboratory - Urine, Pharmacodynamic Blood Sample, ADA, Roll-Over Status, Screening to end of trial, and ECG Results.

The main content area shows the 'Medical History' form for subject BEL9991003. The form includes a 'Medical History' section with a question: 'Has the patient experienced any past clinically significant medical conditions/surgeries or is the patient experiencing any concomitant clinically significant medical conditions?'. The answer is 'Yes'. Below this is a 'Medical History, Log Lines' section with a table of conditions. The table has columns for 'Condition', 'Start date (dd- MMM- yyyy)', and 'Ongoing at Roll-Over visit?'. The first row shows 'dust allergy' with a start date of '12 / 200' and is marked as 'UN' (Unknown). The second row shows 'Start date (dd- MMM- yyyy)' with a start date of '1991' and is marked as 'UNK' (Unknown). The third row shows 'Ongoing at Roll-Over visit?' with a value of '—'. The fourth row shows 'Please specify End date (dd- MMM- yyyy)' with a value of '—'. The interface includes a 'Back To Complete View' button, a 'Previous Line' button, a 'Next Line' button, and an 'Add Another Line' button. A 'Verify' button is also present for each row.

The bottom status bar shows 'Line 2 of 3' and 'Add Another Line'.



ADVANCED **CLINICAL** **RESEARCH** SOLUTIONS

LIFE INSPIRED, QUALITY DRIVEN

CONTACT **US**

 clinicalresearch@sgs.com

 EUROPE: +32 15 27 32 45
AMERICAS: +1 877 677 2667

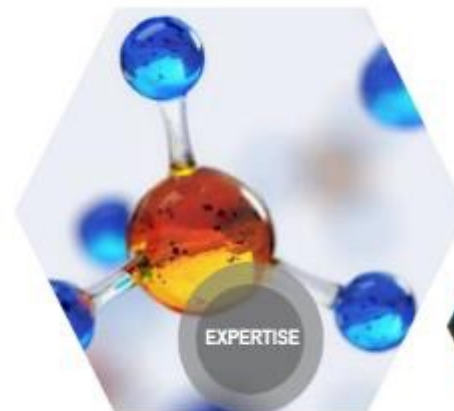
 www.sgs.com/cro



JOIN THE SCIENTIFIC COMMUNITY

Discover R&D market news and events
www.sgs.com/LinkedIn-Life





WWW.SGS.COM

WHEN YOU NEED TO BE SURE

SGS