DYNAMICS REPORT MANAGER
Application Programming Interface

The process for printing a DRM-based report from a third-party dictionary involves these steps.

1. Initializing the Report.
    - Integration_Initialize_Report
2. Setting Parameters.
    - Integration_Parameter_Values_Clear
    - Integration_Parameter_Values_Get
    - Integration_Parameter_Values_Set
    - Integration_Parameter_Values_Add
3. Updating optional actions
    - Integration_Initialize_Email
    - Integration_Email_Add_Address
    - Integration_Update_PDF_Filename
4. Launching the Report.
    - Integration_Launch_Report
    - Integration_Set_Printer
    - Integration_Set_Copy_Count

The basic process flow is to instantiate an Instance of a Report with the Integration_Initialize_Report procedure, which returns an InstanceID. Using the InstanceID, you can set up the rest of the report's parameters and functionality and then launch it.

The Product ID for Dynamics Report Manager is 5014.

Initializing a Report

The Integration_Initialize_Report script is important for clearing the parameter buffers and then setting up default information for the report as well as default parameter values. If all of your parameters are defined to have default values, then it is not necessary to call the parameter setting procedures.

There are two parameters:

1. ReportID. The string ID used in DRM Report Maintenance to uniquely identify a report.
2. InstanceID. The out integer parameter of the report you just instantiated.
3. ErrNum. An error returned by DRM. 0 if OKAY, 1 if Report doesn't exist.

```
{Integration_Initialize_Report}
in string ReportID;
inout integer InstanceID;
out integer ErrNum;
```

An example:

```
local integer InstanceID;
call with name "Integration_Initialize_Report" in dictionary 5014, “SOPDOC”, InstanceID, ErrNum.
if ErrNum <> OKAY then
        warning "An error occurred when calling Integration_Initialize_Report in the DRM.”
end if.
```

Setting Parameters

After the report has been initialized, it is not necessary to set parameter values for those which default. If a default parameter value does exist, these calls will overwrite them.

Integration_Parameter_Values_Clear clears all parameter values for a specific parameter in the report. There are two parameters:

1. InstanceID.
2. ParameterNumber. 1-based integer representing the parameter number to clear.

```
{Integration_Parameter_Values_Clear}
in integer InstanceID;
in integer ParamSeq;
```

An example, assuming the InstanceID is 1 and you want to clear the third parameter's values:

```
call with name "Integration_Parameter_Values_Clear" in dictionary 5014, 1, 3;
```

Additional scripts are listed here work similarly. The integer ValueSeq works like an array for mutli-valued parameters. For single-valued parameters, this value will be 1. Value 1 and Value 2 are used for ranges, with Value1 being the begin of a range and Value2 being the end. For non-range values, Value1 holds the value.

```
{Integration_Parameter_Value_Get}
in integer InstanceID;
in integer ParamSeq;
in integer ValueSeq;
out string Value1;
out string Value2;
```

```
{Integration_Parameter_Value_Set}
in integer InstanceID;
in integer ParamSeq;
in integer ValueSeq;
in string Value1;
in string Value2;
```

If you do not know what values exist already for a parameter, and you wish to add some additional values, you can use this call, which will return the ValueSeq variable telling you which sequence number it used.

```
{Integration_Parameter_Value_Add}
in integer InstanceID;
in integer ParamSeq;
out integer ValueSeq;
in string Value1;
in string Value2;
```

Setting Additional Options

If you are using e-mail, you can use these calls to set up emailing of a report. For the given report instance, you can specify a Subject and a Message for the e-mail. When sent, the report will be attached to the e-mail. You must specify the report to print to pdf.

```
{Integration_Initialize_Email}
in integer InstanceID;
in string Subject;
in text Message;
out integer ErrNum;
```

Use this call to add specific recipients to the e-mail. This call must be made after Integration_Initialize_Email.

```
{Integration_Email_Add_Address}
in integer InstanceID;
in string EmailName;
in string EmailAddress;
out integer ErrNum;
```

Sometimes a PDF document filename is based on the parameters that have been set. You can update the PDF export filename after setting parameters by using this call.

```
{Integration_Update_PDF_Filename}
in integer InstanceID;
```

Launching the Report

Once the parameters are set up and optional calls have been made, you can use this procedure to actually launch the report.

```
{Integration_Launch_Report}
in integer InstanceID;
out integer ErrNum;
```

Use this call to change a printer when launching a report. It would be called after initializing a given report to print, but before launching it.

```
{Integration_Set_Printer}
in integer InstanceID;
in string PrinterID;
```

Use this call to change the number of copies when launching a report. It would be called after initializing a given report to print, but before launching it.

```
{Integration_Set_Copy_Count}
in integer InstanceID;
in string CopyCount;
```

If you have any questions about integrating with DRM, please contact support@rocktonsoftware.com.