Coresuite Service

1. Overview

Coresuite Service is a tool that allows the execution of scheduled rules in SAP Business One databases, independently of a running SAP Business One client. This tool is implemented as a Windows Service.

The new Coresuite Service replaces *Coresuite Customize Service* and has been extended to include the functionality often referred to as *Designer as a Service.*

You can configure Coresuite Service to execute rules on multiple company databases.

For Coresuite Service only the initial configuration for the Service needs to be performed outside of SAP Business One. Any rules are set up from within Coresuite Customize and can therefore easily be shared between databases and environments using the standard Coresuite Customize import and export functionality.

2. Licencing

Coresuite Service requires an additional, separate licence: The Coresuite Service licence.

The minimum licence requirements are outlined below.

General:

- *Coresuite Service* licence per Licence Server (respectively per SAP Business One licence file).

SAP Business One user defined for executing Coresuite Customize Service rules:

- *SAP Business One Indirect Access User* licence.

3. Prerequisites

The tool requires the installation of the Coresuite addon. The minimum Coresuite Installer version is 6.70, the minimum Coresuite Designer and Coresuite Customize version is 6.75. Please refer to our IMPORTANT ANNOUNCEMENT regarding Coresuite Installer and Framework version 6.70 or higher FAQ for relevant details.

Ensure that you have at least 9.3 PL03 installed – previous Patchlevels are NOT supported.

4. Installation and initial Configuration

The initial installation and configuration include the following steps, which are described in individual sections in this document.

We recommend complying with the order of the outlined steps to ensure a successful setup.

1. *Installation of the coresuite addon*
2. *Installation of Coresuite Service*
3. *Configuration of Coresuite Service*
4. *Installation of Coresuite Service licence*
5. *Setting up rules for Coresuite Service*
6. *Starting Coresuite Service*

4. Installation and initial Configuration (continued)

1. Installation of the coresuite addon

New environment

When setting up Coresuite Service in a new environment, perform the basic coresuite addon installation using the Coresuite Installer version 7.60.

Existing environment

When setting up Coresuite Service in an existing environment, ensure you perform (or have already performed) a Coresuite Installer upgrade as per IMPORTANT ANNOUNCEMENT regarding Coresuite Installer and Framework version 6.70 or higher FAQ. When using Coresuite Installer 6.70, you need to upgrade the Coresuite Designer and Coresuite Customize module sips to version 6.75.

⚠ Please note, Coresuite Service is using separate dlls to Coresuite Customize and Coresuite Designer, but is working with these modules which are installed on the database. Due to this, their versions should be aligned (please check the respective Release Notes).

2. Installation of Coresuite Service

It is important to uninstall *Coresuite Customize Service* before installing *Coresuite Service*.

To install the Coresuite Service, run the setup file named "CoresuiteServiceSetup.msi" and follow the instructions on screen. You can decide based on your usage and requirements if you prefer the 32bit or 64bit edition. However, only one edition can be installed on one OS environment at a time.

When installing/upgrading *Coresuite Service*, the current/older version/edition is automatically uninstalled/replaced by the later version/edition.

Please DO NOT start the newly installed service until the configuration has been completed!

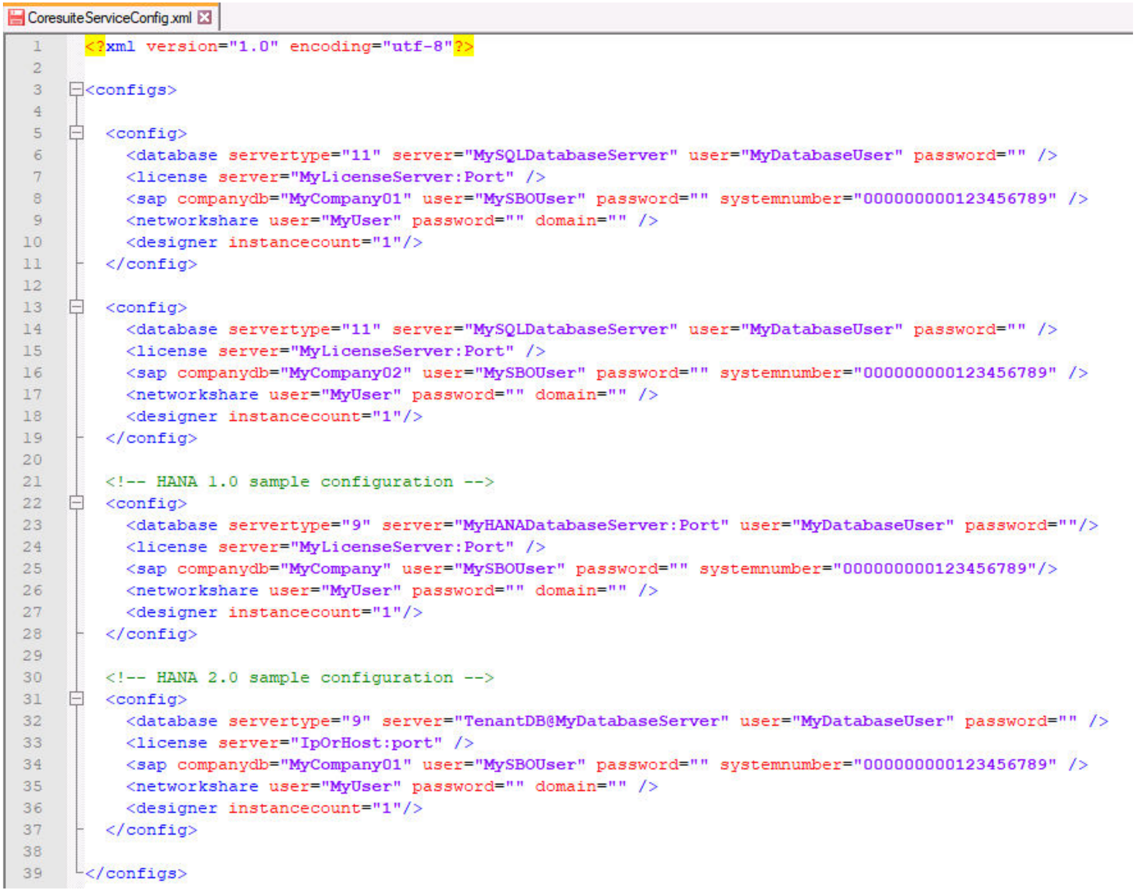
3. Configuration of Coresuite Service

To configure the Service, you need to modify the *CoresuiteServiceConfig.xml*.

1. Go to %ProgramData%\Coresystems\Coresuite Service.
   There are two xml files located directly in that folder. CoresuiteServiceConfig.xml which is the file you need for the configuration, and CoresuiteServiceConfig-sample.xml which is simply a duplicate which facilitates easier reconfiguration if required.

2. Start a simple text or code editor tool via right-click->Run as Administrator and open the CoresuiteServiceConfig.xml. Unless you use the Run as Administrator, you might not be able to save your changes to the file.

3. Update the content with your details:



```xml
<?xml version="1.0" encoding="utf-8"?>

<configs>

  <config>
    <database servertype="11" server="MySQLDatabaseServer" user="MyDatabaseUser" password="" />
    <license server="MyLicenseServer:Port" />
    <sap companydb="MyCompany01" user="MySBOUser" password="" systemnumber="000000000123456789" />
    <networkshare user="MyUser" password="" domain="" />
    <designer instancecount="1"/>
  </config>

  <config>
    <database servertype="11" server="MySQLDatabaseServer" user="MyDatabaseUser" password="" />
    <license server="MyLicenseServer:Port" />
    <sap companydb="MyCompany02" user="MySBOUser" password="" systemnumber="000000000123456789" />
    <networkshare user="MyUser" password="" domain="" />
    <designer instancecount="1"/>
  </config>

  <!-- HANA 1.0 sample configuration -->
  <config>
    <database servertype="9" server="MyHANADatabaseServer:Port" user="MyDatabaseUser" password=""/>
    <license server="MyLicenseServer:Port" />
    <sap companydb="MyCompany" user="MySBOUser" password="" systemnumber="000000000123456789"/>
    <networkshare user="MyUser" password="" domain="" />
    <designer instancecount="1"/>
  </config>

  <!-- HANA 2.0 sample configuration -->
  <config>
    <database servertype="9" server="TenantDB@MyDatabaseServer" user="MyDatabaseUser" password="" />
    <license server="IpOrHost:port" />
    <sap companydb="MyCompany01" user="MySBOUser" password="" systemnumber="000000000123456789" />
    <networkshare user="MyUser" password="" domain="" />
    <designer instancecount="1"/>
  </config>

</configs>
```

Image 01 – CoresuiteSrviceConfig.xml

- **(database) servertype:**
  Specify your server type according to the key provided in the file *e.g. enter 11 when using MS SQL Server 2017*

- **(database) server**:
  Specify the name of the server. For SQL servers, the actual server name or IP address of the server is sufficient, for HANA servers, the port must also be provided e.g. 192.168.2.1:30015
  MSSQL: server="MyDatabaseServer"
  HANA 1.0: server="MyHANADatabaseServer:Port"
  HANA 2.0: server="TenantDB@MyDatabaseServer:Port"

- **(database) user**:
  Specify the name of the user connecting to the database e.g. sa/SYSTEM or an alternative sa/SYSTEM user.

- **(license) server**
  Specify the name/IP address including the port of the license server you are connecting to.
  MSSQL: <license server="MyLicenseServer:Port" />
  HANA 1.0: <license server="MyLicenseServer:Port" />
  HANA 2.0: <license server="IpOrHost:Port" />

- **(sap) companydb**
  Specify the name of your company database.

- **(sap) user:**
  Specify the SAP Business One user ID of the user with whose credentials Coresuite Service should execute the rules.

- **(database / sap) password**
  Specify the password for the above user.
  The value of 'password' attributes will be encrypted when the service is started. The encrypted value is then stored in 'ct' attribute.
  To change a password, simply delete the corresponding 'ct' attribute with its value and add a new 'password' attribute with a new password as its value.
  Please keep in mind that some characters as such are not accepted when they are used in XML file. If your password contains any of characters below, you need to use their escaped version:

"	=	&quot;
‘	=	&apos;
<	=	&alt;
>	=	&gt;
&	=	&amp;

- **(sap) systemnumber**
  Specify the System number from the SAP licence file assigned to the specified company database.

- **networkshare**
  Use this xxx, in case your process needs to access file stored on other servers only.
  The <networkshare user="MyUser" password="" domain="" /> is optional and only required if using any Network Shared Drive.

- **(designer) instancecount**
  This attribute defines the number of maximum running simultaneous Designer instances. It balances a load of document rendering as is showcased in Coresuite Service Rules with the name ‘*SAMPLE: Print Open Sales Orders Load Balancing’*. The more instances are configured, the more HW resources will be required. We recommend running maximum 5 instances and monitoring resource consumption.

Each database needs its own <config> section.

When using only a single database with Coresuite Service, simply remove the excess <config> sections. Add additional <config> sections for each additional database you want to use with Coresuite Service.

Please note that we always provide an up-to-date version of the *CoresuiteServiceConfig-sample.xml* file in in C:\ProgramData\Coresystems\CoresuiteService.

4. Installation of Coresuite Service licence

The Coresuite Service licence is available via our portal. Please request the licence and update your current licence file via the Coresuite Administration Assistant or alternatively via *Administration->Add-ons->coresuite administration->License management* => *Download license* in your respective company database.

Assign this licence to the SAP Business One user defined for executing Coresuite Customize Service rules.

5. Setting up rules for Coresuite Service

Only users defined as “Superuser” can create/modify Coresuite Service Rules. This mechanism has been set up to ensure that “standard” users do not create/modify Coresuite Service Rules by accident/unintentionally.

To create rules, go to *Administration->Add-ons-> coresuite customize-> Coresuite Service Rules.*

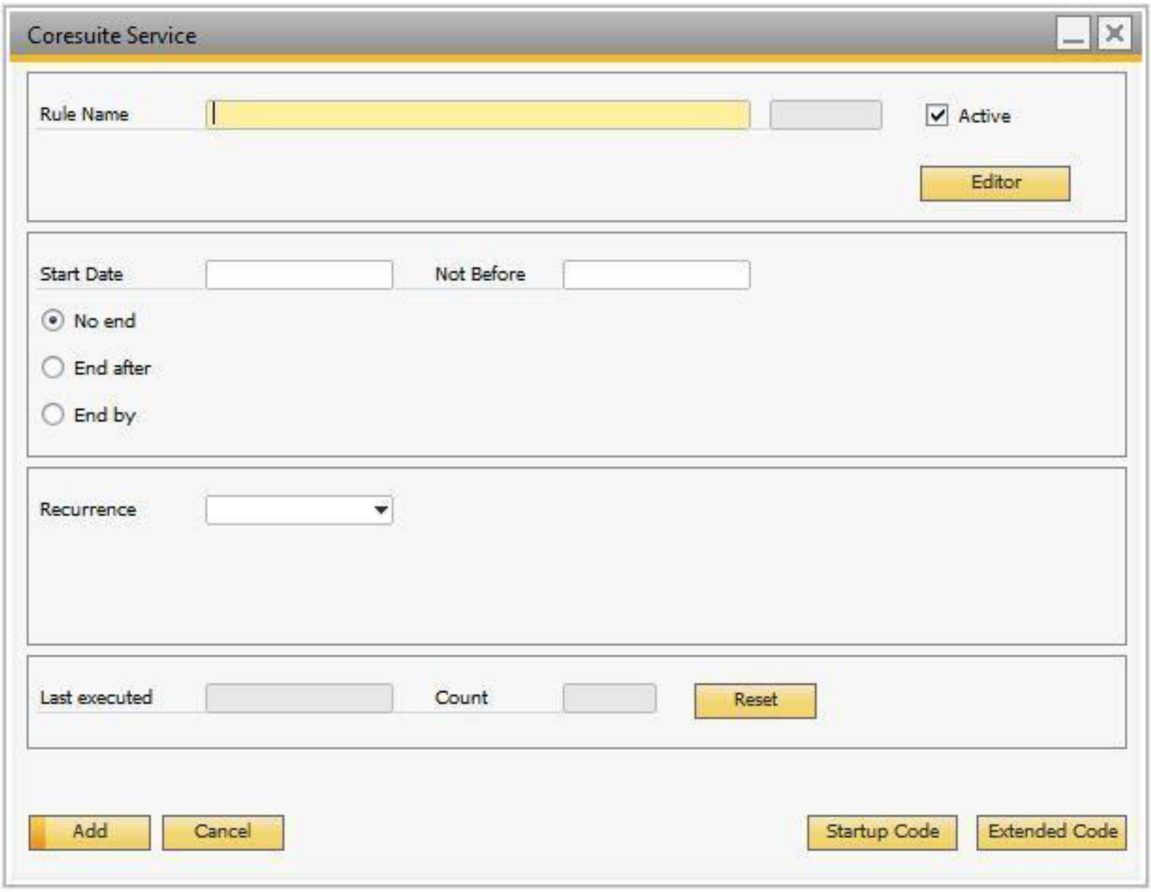


Image 02 – Coresuite Service rule window

In the main window, you can specify the scheduling for the rule and access the Editor to write the code for your rule.

The *Startup Code* and *Extended Code* buttons give access to the Coresuite Service Start Up and Extended Code. Please note that these differ from the general Coresuite Customize Start Up and Extended Code.

Click on the “Editor” button to write your rule.

Coresystems delivers a sample for the printing of open Sales Orders (*SAMPLE: Print Open Sales Orders*) including the Extended Code which is used in the said rule, a sample rule for deleting of old Coresuite Service logs (*SAMPLE: Delete Old Service Logs*), and a sample rule (*SAMPLE: Print Open Sales Orders Load Balancing*) to showcase addressing of multiple Designer instances. Sample Rules and the sample Extended Code are automatically added when installing Coresuite Customize.

⚠ Please check out to the section “Details of provided sample rule”.

To create your first rule, you can simply duplicate the provided sample rule and apply your required modification.

6. Starting Coresuite Service

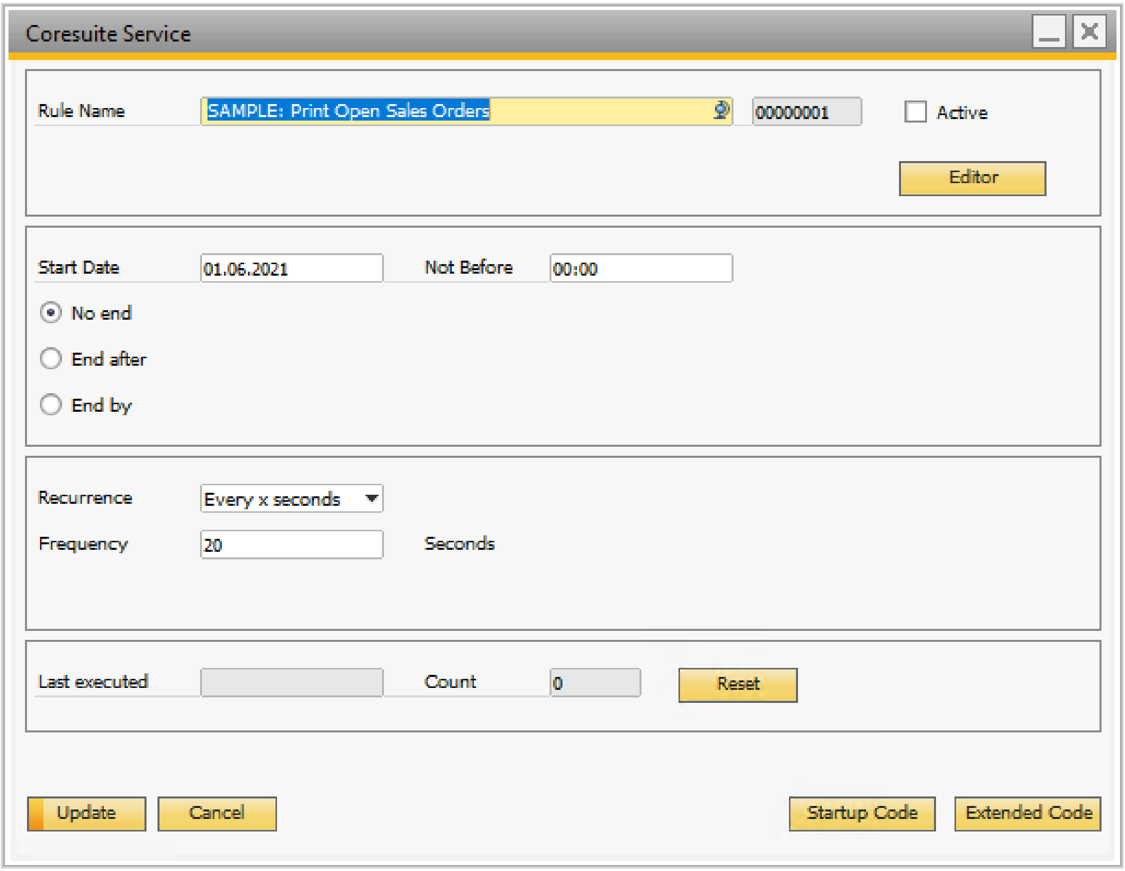
To start Coresuite Service, go to your windows Services where you will now find “Coresuite Service”.

When you start Coresuite Service for the first time, the passwords entered in the CoresuiteServiceConfig.xml file are automatically encrypted.

⚠ Each time you either modify an existing or create a new Coresuite Service rule, Coresuite Service needs to be restarted!

5. Details of provided sample rule “SAMPLE: Print Open Sales Orders”

This sample rule once set to “Active” runs every 20 seconds and creates a PDF for any open and unprinted Sales Orders.



Coresuite Service

Rule Name: SAMPLE: Print Open Sales Orders 00000001 □ Active

Start Date: 01.06.2021 Not Before: 00:00

● No end

○ End after

○ End by

Recurrence: Every x seconds

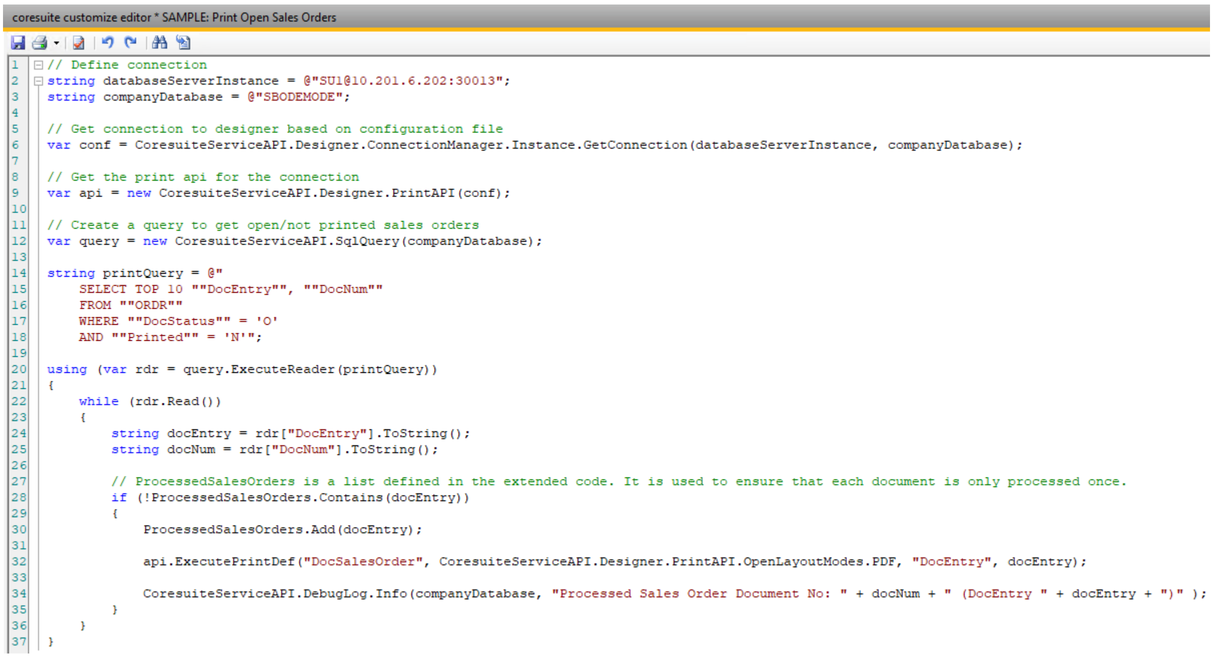
Frequency: 20 Seconds

Last executed: Count: 0

Image 03 – Sample Rule - Main Screen

⚠ Please consider the performance of the database server / application server, the number of users working in the company database, and other customizations when setting Recurrence / Frequency.
We do not recommend configuring an execution Frequency for rules too often as it takes some time to complete the query (depending on your query) and then rendering of documents takes also some time. If it is expected that many documents will be rendered by some rule, please check the *Load Balancing* functionality which is demonstrated in Coresuite Service Rule *SAMPLE: Print Open Sales Orders Load Balancing*.

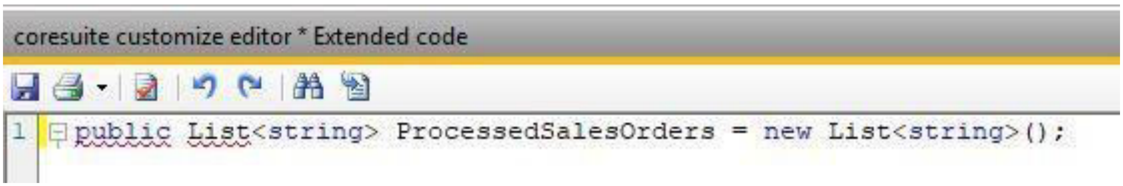
5. Details of provided sample rule (continued)



coresuite customize editor * SAMPLE: Print Open Sales Orders

```
// Define connection
string databaseServerInstance = @"SU1@10.201.6.202:30013";
string companyDatabase = @"SBODEMODE";

// Get connection to designer based on configuration file
var conf = CoresuiteServiceAPI.Designer.ConnectionManager.Instance.GetConnection(databaseServerInstance, companyDatabase);

// Get the print api for the connection
var api = new CoresuiteServiceAPI.Designer.PrintAPI(conf);

// Create a query to get open/not printed sales orders
var query = new CoresuiteServiceAPI.SqlQuery(companyDatabase);

string printQuery = @"
    SELECT TOP 10 ""DocEntry"", ""DocNum""
    FROM ""ORDR""
    WHERE ""DocStatus"" = 'O'
    AND ""Printed"" = 'N'";

using (var rdr = query.ExecuteReader(printQuery))
{
    while (rdr.Read())
    {
        string docEntry = rdr["DocEntry"].ToString();
        string docNum = rdr["DocNum"].ToString();

        // ProcessedSalesOrders is a list defined in the extended code. It is used to ensure that each document is only processed once.
        if (!ProcessedSalesOrders.Contains(docEntry))
        {
            ProcessedSalesOrders.Add(docEntry);

            api.ExecutePrintDef("DocSalesOrder", CoresuiteServiceAPI.Designer.PrintAPI.OpenLayoutModes.PDF, "DocEntry", docEntry);

            CoresuiteServiceAPI.DebugLog.Info(companyDatabase, "Processed Sales Order Document No: " + docNum + " (DocEntry " + docEntry + ")" );
        }
    }
}
```

Image 04 – Sample Rule – Code



```
public List<string> ProcessedSalesOrders = new List<string>();
```

Image 05 – Sample Rule - Extended code

- The Query determines which documents should be processed.
  Depending on the set frequency for the rule execution, using e.g. “TOP 10” in the query can help preventing that a document is processed multiple times.

- Using the List defined in the Extended code is another example how to ensure that the same documents are not processed multiple times unintentionally.

- To specify which Printdefinition has to be used for executing the required printjob, we use the *Name Id* of said Printdefintion (Printdefinition-> Extended Definition-> Name Id).

6. Troubleshooting Help

There are several locations where you can find details logged which should help you to troubleshoot any issues:

- Debug log for the service:
  Please check %ProgramData%\Coresystems\Coresuite Service\DebugLogs

- Debug log for Coresuite Designer and Coresuite Customize used in Coresuite Service:
  Please check C:\Windows\Temp\coresuite [(1)] for log files named *coresuite-%companyDBName%.log* .
  [(1)] for version 7.35 and lower, the path is C:\Windows\Temp\ and log file name is *coresuiteDebugLog.xml*

- Log table within company database:
  Please check "@COR_CUSTOM_SVCLOG"

7. Other functionalities

The Coresuite Designer provides the option to perform *Remote Printing* meaning a third-party tool can create xml files that the Coresuite Designer can read and process.

Coresuite Service facilitates *Remote Printing* without a running SAP Business One Client. There is no further setup required, should you have your system already configured to perform Remote Printing.

For details on Remote Printing, please see check out our Coresuite Country Package Manual.

Report Subscription can be executed by Coresuite Service, so no user has to be logged-in in SAP Business One in order to use the functionality of configured Subscriptions.

You can execute your Print Definitions from external applications (your own modules connecting to Coresuite Designer).

Coresuite Service can be used for emailing via Office 365. For more details related to configuration, please see Emailing-with-Coresuite-Designer-and-Coresuite-Service-via-Office-365.

Rendering of documents can be time-consuming due to layout objects that have to be processed. Coresuite Service can now balance the load via addressing multiple Designer instances, so each of those instances is responsible for rendering different documents.

8. Use cases

In the sample rule, we are simply “printing” open and unprinted Sales Quotation shortly after they were added (hence the high frequency of rule execution).

Here, another simple use case “processing Documents created via the Billing Wizard”:

- You create AR Invoices with the Billing Wizard included in Coresuite Time and you want to send the Document to the Customer but also print it for your own records
- Current Limitation (without the use of further Coresuite Customize rules) would only allow to automatically generate one printjob per AR Invoice using the Default Printdefinition defined for the Billing Wizard

In this scenario you have several options to meet your customers requirement:

- You could simply write a query to check for those AR Invoices and use this query in two Coresuite Service rules.
  In the query you could compare the DocDate with today’s date together with ensuring the Datasource is “O” (see SAP Business One SDK help)
- You could also allow the Billing Wizard to perform the first printjob (sending the email to the customer) and then use a rule to print same documents