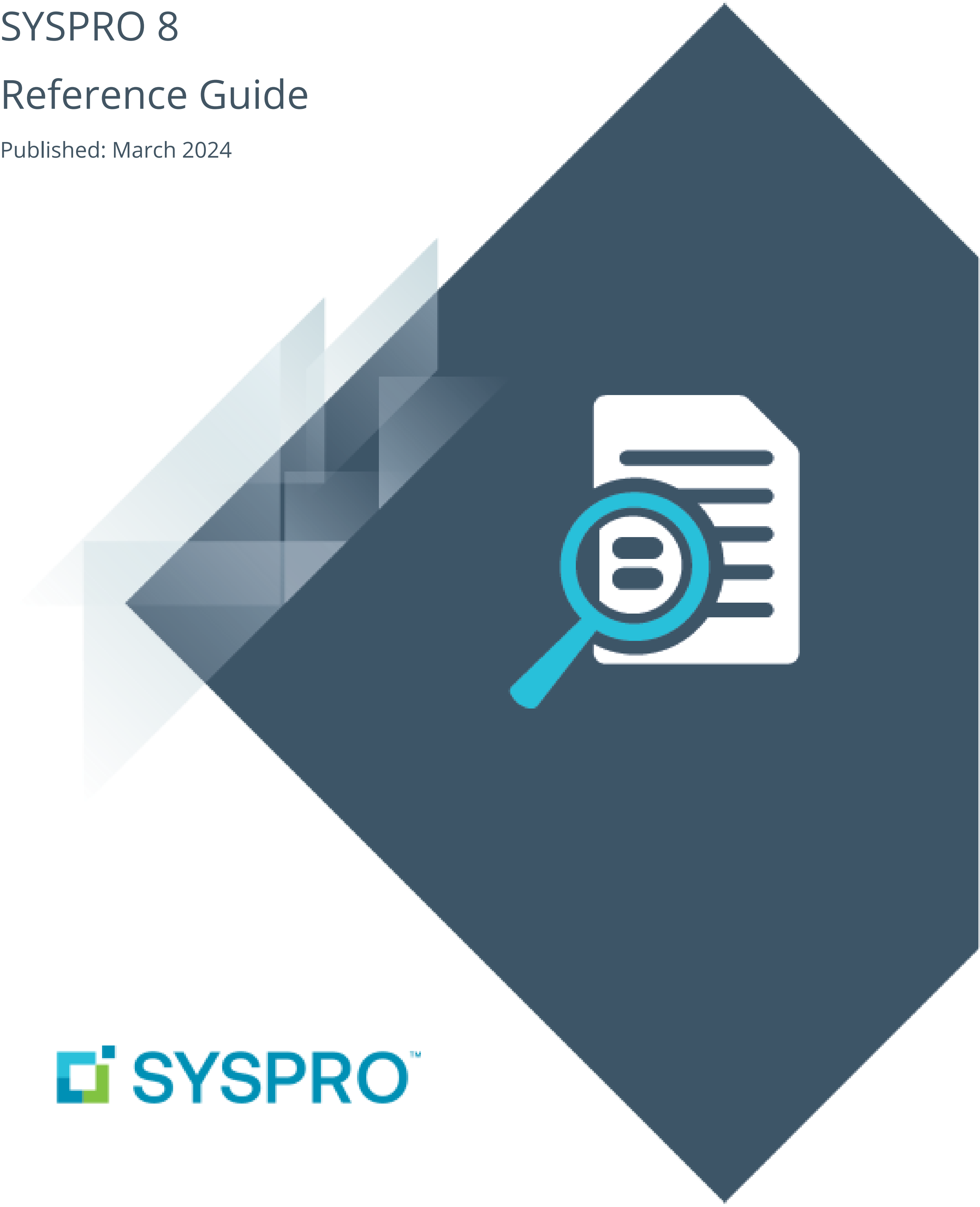
SYSPRO Open Reporting API

SYSPRO 8

Reference Guide

Published: March 2024



CONTENTS

Open Reporting API

Open Reporting API

Exploring

Where it fits in?

The **Open Reporting API** lets developers and external applications call SYSPRO to run and distribute documents directly in the external application.

Leveraging the SYSPRO Reporting Service Server infrastructure, developers can query the SYSPRO database and produce the required documents which are added to the print queue from where they can be viewed, executed and managed. Developers can access the document via the document queue for further automation (a destination code indicates the origin of the queue item).

Functionality

The **Open Reporting API** enables custom application or third party developers to programmatically produce single documents from applications outside the main SYSPRO application, by referencing an assembly located within the SYSPRO `\Base` folder.

The **Open Reporting API** works by creating a business object wrapper around the standard SYSPRO print programs and by providing business objects to retrieve the required information about those documents that are printed. Please refer to the **Referencing** section for more information on the XML that is used for each document type.

Documents that have already been generated and stored in the document queue can't be retrieved via the API.

Each document type first needs to be converted to support the **Open Reporting API** before the wrapper business objects can be written to call them. This is to ensure that the SYSPRO print programs don't try to show any user interface when called by the API. As a result, not all document types may be supported by the API.

Disclaimer:

The business objects used by the **Open Reporting API** are intended for use within the API only and should not be used directly in e.net. SYSPRO reserves the right to change the way these business objects work in order to support the API.

Starting

Prerequisites

Before using the **Open Reporting API**, ensure that **Server-side Printing** is configured correctly in SYSPRO.

The following is required:

Server Technology Requirements

- Reporting Host Service
- Crystal Reports Server Embedded (CR 2016 and any specific service packs)
- SAP BusinessObjects BI platform .Net SDK Redistributable 64-bit 4.2. SP3

Client Configuration Requirements

- Browser pop-ups must be enabled

> This is only applicable for previewing PDFs in a web-browser application.

SYSPRO Configuration Requirements

- Server-side reporting (configured within the SYSPRO **Setup Options** program *Setup Options* > *System Setup* > *Reporting*).
- e.Net Service Details (configured within the SYSPRO **Setup Options** program *Setup Options* > *System Setup* > *Artificial Intelligence*).
- SMTP emailing (configured within the SYSPRO **Setup Options** program *Setup Options* > *System Setup* > *Connectivity*).

Restrictions and Limits

- The **Open Reporting API** is only supported for SRS server-side printing, as it makes use of the **SYSPRO 8 Reporting Host Service** to call the print business objects and generate documents.

> The **Open Reporting API** is therefore not available for client-side printing.

- When emailing a document, only one email address can be entered in the **To** and **CC** lines.

- Known limitations:

  The following are known limitations, which may be addressed during the lifetime of ***SYSPRO 8***:

  - Batch printing of documents is currently not supported.
  - Report printing is not supported.
  - Documents that are printed using this architecture are currently not added to the document archive.

    The generated documents are stored in the SRS document queue used for server side printing and will be visible from the **SRS Document Queue** program. However, they are not added to the document archive.

    There will be no PDF file in the **SRS Document Printing** archive folder, nor will the document be visible in the **Document Archive Viewer**.
  - This functionality is limited to the following document types:
    - Sales order documents
      - Invoices
      - Delivery notes
      - Order acknowledgments
      - Dispatch notes
      - Dispatch note invoices
    - AR statements
    - Quotations
    - Factory documentation
    - Foreign and local purchase orders (excluding requisitions)

    > Support for other documents and reports may be made available in the future.

Configuring

Setup Options

The **Setup Options** program lets you configure how SYSPRO behaves across all modules. These settings can affect processing within this program.

E.Net Service Details System Setup

Setup Options > System Setup > E.Net Service Details

- Server name
- SOAP port
- REST port

Reporting System Setup

Setup Options > System Setup > Reporting

- Reporting configuration
- Server-side configuration

Solving

FAQs

Where can I download the SYSPROSRSClientLibrary.dll?

The `SYSPROSRSClientLibrary.dll` assembly file is located in your SYSPRO `\Base` folder.

It is placed into this folder during the installation of ***SYSPRO 8***.

Where can I find the SYSPROSRSDocumentAPI.dll assembly?

This file is no longer in use and has been replaced by the `SYSPROSRSClientLibrary.dll` assembly file.

The `SYSPROSRSClientLibrary.dll` assembly (located in your `\Base` directory) is used by all applications that want to use the functionality of the SRS API.

> All DLLs are located in your `\Base` directory.

Where can I see the host service end-point in SYSPRO?

1. Load the **System Setup** program (*SYSPRO Ribbon bar* > *Setup* > *General Setup*).
2. Navigate to the **Reporting** tab.
3. The host service end-point is displayed at the **REPORTING SERVICE** field of the **SERVER-SIDE CONFIGURATION** section.

Which assembly should be used for which functionality?

1. The `SYSPROSRSClientLibrary.dll` assembly is used by all applications that want to use the printing or reporting services functionality.
2. The `SysproWCFClientLibrary40.dll` assembly can be used for all applications that want to access SYSPRO.

   > The `SysproWCFClientLibrary40.dll` is not a requirement for printing.

Could the API be used for SWS to email the document?

Yes, you can produce a document from within SYSPRO workflow if you create a custom activity that will allow you to create a reference to the client library assembly and write your C# code.

Using

Process

Installing/starting the SRS Host Service

The **SYSPRO 8 Reporting Host Service** is deployed via the **SYSPRO Installer** and can be installed automatically when ***SYSPRO 8*** is installed.

We recommend stopping the **SYSPRO 8 Reporting Host Service** before any updates are done and restarting it thereafter.

> You only need to reinstall the service if an updated and/or improved version of the service is released.

Configuring server-side reporting

1. From the **Setup Options** program, select the **Reporting System Setup** form (*Setup Options > System Setup > Reporting*).
2. Configure the required settings:

Field	Action
Reporting configuration	Select **Server-side reporting using SQL**.
SQL Server Name	Enter the name of the SQL Server instance that contains the _ `SRS` database.
Reporting authentication	Select the type of authentication you will use.

3. Save your changes.

   Ensure that you can successfully print the document type using SYSPRO before trying to print it using the API.

How to use the API

All API calls are done via the SRS library.

1. Create a new project and add the `DLL` as a reference.
2. Use the five available methods to query and generate documents.

How to add a reference to the client library assembly

1. Open the solution.
2. Select the **Solution Explorer** (*View > Solution Explorer*).
3. Right-click on **References** in the **Solution Explorer** pane of the project and select **Add Reference**.
4. Select **Browse** and navigate to the SYSPRO `\Base` folder.
5. Select the reference, e.g. `SYSPROSRSClientLibrary.dll` and click on **Add**.
6. Ensure the assembly is checked in the list of assemblies and select **OK**.

Affected business objects

The following indicates the business objects that are affected by this feature:

Query objects

SO Document Details Query

The **SO DOCUMENT DETAILS QUERY**[1] business object is used by the Espresso Application (or third party program) to obtain sales order and invoice details. This then returns a list of documents that can be printed.

SO Document Print Query

The **SO DOCUMENT PRINT QUERY**[2] business object is used by the Espresso Application (or third party program) to obtain and return information in the XML format required to generate SRS documents.

[1] Business object: SORQD2

[2] Business object: SORQDP

Referencing

API Methods

The exposed API methods can be used in a number of ways:

- Authentication.
- Controlling login and logout.
- Controlling the formats of the documents.

This lets customers with login credentials access their data to generate documentation from third-party applications. External or third-party applications like ***SYSPRO Espresso*** can request the information from the API and, once received, apply it to generate the required documents.

The `SYSPROSRSClientLibrary.dll` assembly (located in the `\Base` folder) allows an application like ***SYSPRO Espresso*** to call the **SYSPRO 8 Reporting Host Service** in order to create the documents.

The following API methods or functions are exposed:

AuthSysproUser

This controls and authorizes the logging on of users. This method returns a session ID.

> Because the API logs on via the service, which is a trusted application, SYSPRO doesn't require a password. Although the method requests passwords, these aren't actually used.
>
> This means that you need to make sure that any operator log on authentication is handled by the third party or custom application.

AuthenticateSysproGUID

This controls and authorizes access of users that are already logged in. This method returns a session ID.

> This takes a session ID you have generated by doing a logon using the **SYSPRO 8 e.net Communications Load Balancer** and authenticates that as a valid session ID and returns a GUID.

LogOffUserSession

This logs off sessions of stand-alone applications.

> This takes the GUID that is returned by the `AuthSysproUser` function and performs a log off.
>
> This GUID that is returned is not the same as the session ID that is returned from a log on.

DetermineDocument Options

This provides a list of formats for the selected document type as well as sometimes additional information pertaining to the selected document. This method returns XML with details of available documents and formats.

> This only returns SRS formats and looks at the control file (i.e. sales order control file), to determine which formats are defined for SRS according to the governing business rules.

This method returns the following that can be used to determine which document types can be printed:

- For the **Sales Order Document Printing** functionality:
  - Sales order number
  - Sales order status
  - Sales order flags
  - Dispatch note number
  - Print & reprint flags
- For other document types:
  - Purchase order number
  - Quote number
  - Factory documentation number

ProduceDocument

This generates the document to print on the host server.

When previewing the document, it is returned in a HEX encoded format of a PDF document

This content must be converted back into ASCII format, before saving it to disk.

The input XML is passed to the business object, which in turn communicates with the **Document Print** program. The **SYSPRO 8 Reporting Host Service** uses the XML returned from the **Document Print** program, to generate the document using the selected format.

Sample XML per document type

Determine document options

API method name: Determine document options

Sales Orders

This function can be used to return information on a specific sales order as well as the formats that have been configured per sales order document type. This allows the application developer to present a user interface to their customer and control which documents can be printed and which print options are available.

For example, if both the `CanPrintAcknowledgement` and `CanRePrintAcknowledgement` values are `FALSE`, then no order acknowledgment can be generated or reprinted for the sales order.

Information for any invoices that have been generated for a sales order will be returned and if there are any lines available to invoice on the sales order then the `CanGenerateInvoice` element will have a value of `TRUE`.

Various document types will return flags to indicate whether a document can be printed or reprinted. It is up to the 3rd party application to read and store these flags and then provide the correct value for `Reprint` element in the produce document input XML ,where applicable.

- The following codes are used for the sales order documents types:
  - `I` - invoice
  - `D` - delivery note
  - `O` - order acknowledgment
  - `N` - dispatch notes

XML In

```
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Option>
    <Function>GETDOCDET</Function>
  </Option>
  <Filter>
    <OrderNumber FilterType='S' FilterValue='791' />
  </Filter>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl
Language="05" Language2="EN" CssStyle="" DecFormat="1" DateFormat="01" Role="01" Version="8.0.001" OperatorPrimaryRole="
">
  <DocumentInformation>
    <SalesOrder>000791</SalesOrder>
    <OrderStatus>9</OrderStatus>
    <Translated_OrderStatus>Complete</Translated_OrderStatus>
```

```
      <ActiveFlag>N</ActiveFlag>
      <Translated_ActiveFlag>No</Translated_ActiveFlag>
      <CancelledFlag />
      <Translated_CancelledFlag>No</Translated_CancelledFlag>
      <SODocumentType>O</SODocumentType>
      <Translated_SODocumentType>Order</Translated_SODocumentType>
      <CanPrintAcknowledgement>false</CanPrintAcknowledgement>
      <CanRePrintAcknowledgement>false</CanRePrintAcknowledgement>
      <CanPrintDeliveryNote>false</CanPrintDeliveryNote>
      <CanRePrintDeliveryNote>false</CanRePrintDeliveryNote>
      <CanGenerateInvoice>false</CanGenerateInvoice>
      <Documents>
        <Invoice>
          <InvoiceNumber>100506</InvoiceNumber>
          <InvoiceSource>O</InvoiceSource>
          <Translated_InvoiceSource>Order</Translated_InvoiceSource>
          <DateLastInvPrt>2015-04-08</DateLastInvPrt>
        </Invoice>
      </Documents>
      <Formats>
        <Format>
          <DocumentType>Order Acknowledgement</DocumentType>
          <FormatCode>0</FormatCode>
          <FormatName>Order Acknowledgemen</FormatName>
        </Format>
        <Format>
          <DocumentType>Delivery Note</DocumentType>
          <FormatCode>0</FormatCode>
          <FormatName>Delivery Note</FormatName>
        </Format>
        <Format>
          <DocumentType>Invoice</DocumentType>
          <FormatCode>0</FormatCode>
          <FormatName>Invoice</FormatName>
        </Format>
      </Formats>
    </DocumentInformation>
  </DocumentControl>
```

Accounts Receivable statement print

The only options available for this document type is to return a list of the SRS formats that have been configured.

XML In

```
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Option>
    <Function>GETFMTS</Function>
    <Format />
  </Option>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="Windows-1252"?>
<DocumentControl
Language="05" Language2="EN" CssStyle="" DecFormat="1" DateFormat="01" Role="01" Version="8.0.001" OperatorPrimaryRole="
">
  <DocumentInformation>
    <DocumentType>A/R Statement Print</DocumentType>
    <Format>
      <FormatCode>01</FormatCode>
      <FormatName>Customer Statement</FormatName>
    </Format>
  </DocumentInformation>
</DocumentControl>
```

Purchase orders (foreign and local)

This function returns information for a given purchase order as well as all the formats that have been defined so that the user interface can prompt the user to select a format and print. As with the sales order document type, flags are returned to indicate whether the document can be printed and reprinted.

XML In

```xml
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>GETDOCDET</Function>
    <IncludeFormatDetails>Y</IncludeFormatDetails>
  </Option>
  <Filter>
    <PurchaseOrder FilterType="S" FilterValue="420" />
  </Filter>
</Query>
```

XML Out

```xml
<?xml version="1.0" encoding="Windows-1252"?>
<DocumentControl
Language="05" Language2="EN" CssStyle="" DecFormat="1" DateFormat="01" Role="01" Version="8.0.000" OperatorPrimaryRole="
">
  <DocumentInformation>
    <PurchaseOrder>000420</PurchaseOrder>
    <PurchaseOrderStatus>4</PurchaseOrderStatus>
    <Translated_OrderStatus>Order printed</Translated_OrderStatus>
    <PurchaseOrderType>I</PurchaseOrderType>
    <Translated_PurchaseOrderType>Import order</Translated_PurchaseOrderType>
    <PurchaseOrderActiveFlag />
    <PurchaseOrderCancelledFlag />
    <CanPrintPurchaseOrder>false</CanPrintPurchaseOrder>
    <CanRePrintPurchaseOrder>true</CanRePrintPurchaseOrder>
    <Formats>
      <Format>
        <DocumentType>F</DocumentType>
        <DocumentTypeDescription>Purchase orders - foreign</DocumentTypeDescription>
        <FormatCode>0</FormatCode>
        <FormatName>P/order - Foreign</FormatName>
      </Format>
      <Format>
        <DocumentType>L</DocumentType>
        <DocumentTypeDescription>Purchase orders - Local</DocumentTypeDescription>
        <FormatCode>0</FormatCode>
        <FormatName>P/order - Local</FormatName>
      </Format>
    </Formats>
  </DocumentInformation>
</DocumentControl>
```

Quotations

This function returns information for a given quotation as well as all the formats that have been defined so that the user interface can prompt the user to select a format and print. As with the sales order document type, flags are returned to indicate whether the document can be printed and reprinted.

XML In

```
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>GETDOCDET</Function>
    <IncludeFormatDetails>Y</IncludeFormatDetails>
  </Option>
  <Filter>
    <Quotation FilterType="S" FilterValue="344" />
  </Filter>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="Windows-1252"?>
<DocumentControl
Language="05" Language2="EN" CssStyle="" DecFormat="1" DateFormat="01" Role="01" Version="8.0.001" OperatorPrimaryRole="
">
  <DocumentInformation>
    <Quotation>00000344</Quotation>
    <QuotationVersion>000</QuotationVersion>
    <QuoteStatus>0</QuoteStatus>
    <Translated_OrderStatus>In progress</Translated_OrderStatus>
    <CanPrintQuotation>false</CanPrintQuotation>
    <CanRePrintQuotation>false</CanRePrintQuotation>
    <Formats>
      <Format>
        <FormatCode>0</FormatCode>
        <FormatName>Quotation</FormatName>
      </Format>
      <Format>
        <FormatCode>1</FormatCode>
        <FormatName>Quotation - Multiple</FormatName>
      </Format>
    </Formats>
  </DocumentInformation>
</DocumentControl>
```

Factory documentation

The only options available for this document type is to return a list of the SRS formats that have been configured.

XML In

```
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>GETFMTS</Function>
  </Option>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="Windows-1252"?>
<DocumentControl
Language="05" Language2="EN" CssStyle="" DecFormat="1" DateFormat="01" Role="01" Version="8.0.001" OperatorPrimaryRole="
">
  <DocumentInformation>
    <Format>
      <DocumentNumber>1</DocumentNumber>
      <FormatCode>1</FormatCode>
      <FormatName>Job Card</FormatName>
    </Format>
    <Format>
      <DocumentNumber>2</DocumentNumber>
      <FormatCode>3</FormatCode>
      <FormatName>Picking Slip</FormatName>
    </Format>
  </DocumentInformation>
</DocumentControl>
```

Produce document

API method name: ProduceDocument

The XML parameters for each document type are the same, the only difference would be the `DocumentType` element which indicates for which document type you are generating the document. The `DocumentType` element is not case sensitive.

The purpose of the parameter XML is to tell the function what you would like to do with the document that is generated.

Sales Orders

XML Parameters

```
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl>
  <DocumentType>SalesOrder</DocumentType>
  <Print>False</Print>
  <Email>False</Email>
  <Preview>True</Preview>
  <XmlOnly>False</XmlOnly>
  <PrinterDetails>
    <PrinterName />
    <PrintCopies>1</PrintCopies>
    <PrintCollate>True</PrintCollate>
  </PrinterDetails>
  <EmailDetails>
    <EmailFromAddress />
    <EmailToAddress />
    <EmailCCAddress />
    <EmailToAddress />
    <EmailBodyText />
  </EmailDetails>
</DocumentControl>
```

XML In

When printing a sales order, you need to supply the `DocumentType` element to tell the API which sales order document you want to produce, in this case it is an invoice.

> **FOR EXAMPLE:**
>
> This instructs the API to reprint invoice 100506 for sales order 791 using format 0.

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Option>
    <Function>ONLINE</Function>
    <DocumentType>I</DocumentType>
    <Format>0</Format>
    <Reprint>Y</Reprint>
    <PrintDispatchInvoice>N</PrintDispatchInvoice>
  </Option>
  <Filter>
    <OrderNumber FilterType='S' FilterValue='000791' />
    <InvoiceNumber FilterType='S' FilterValue='100506' />
  </Filter>
</Query>
```

> **FOR EXAMPLE:**
>
> This instructs the API to reprint dispatch note invoice 100540 for sales order 945 and dispatch note 0000029 using format 0.

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Option>
    <Function>ONLINE</Function>
    <DocumentType>I</DocumentType>
    <Format>0</Format>
    <Reprint>Y</Reprint>
    <PrintDispatchInvoice>Y</PrintDispatchInvoice>
  </Option>
  <Filter>
    <InvoiceNumber FilterType="S" FilterValue="100540" />
    <OrderNumber FilterType="S" FilterValue="000000000000945" />
    <DispatchNote FilterType="S" FilterValue="000000000000027" />
  </Filter>
</Query>
```

XML Out

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Document>
    <Status />
    <DocumentGuid/>
    <ErrorMessage/>
    <StatusPreview/>
    <DocumentHex>HEX Encoded PDF file</DocumentHex>
  </Document>
</Query>
```

Accounts Receivable statement print

XML Parameters

```
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl>
  <DocumentType>AR Statement Print</DocumentType>
  <Print>False</Print>
  <Email>False</Email>
  <Preview>True</Preview>
  <XmlOnly>False</XmlOnly>
  <PrinterDetails>
    <PrinterName />
    <PrintCopies>1</PrintCopies>
    <PrintCollate>True</PrintCollate>
  </PrinterDetails>
  <EmailDetails>
    <EmailFromAddress />
    <EmailToAddress />
    <EmailCCAddress />
    <EmailToAddress />
    <EmailBodyText />
  </EmailDetails>
</DocumentControl>
```

XML In

The AR Statement format allows for multiple options to be provided via a 3rd party API.

For more information on these elements, please refer to the **AR Statement Print** program in SYSPRO

```
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>ONLINE</Function>
    <Format>0</Format>
    <StatementAsOf>C</StatementAsOf>
    <StatementDate />
    <StatementAgeing>S</StatementAgeing>
    <ConsolidateSub>N</ConsolidateSub>
    <IncludeAttached>N</IncludeAttached>
    <AsOpenItem>N</AsOpenItem>
    <BalanceType>A</BalanceType>
    <MinimumBalance>0.00</MinimumBalance>
    <SalesMessage />
  </Option>
  <Filter>
    <Customer FilterType="S" FilterValue="0000001" />
  </Filter>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Document>
    <Status />
    <DocumentGuid/>
    <ErrorMessage/>
    <StatusPreview/>
    <DocumentHex>HEX Encoded PDF file</DocumentHex>
  </Document>
</Query>
```

Purchase orders (foreign and lcoal)

XML Parameters

Supported values for the `DocumentType` element are `Purchase orders - Local` and `Purchase orders - Foreign`.

```
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl>
  <DocumentType>Purchase orders - Local</DocumentType>
  <Print>False</Print>
  <Email>False</Email>
  <Preview>True</Preview>
  <XmlOnly>False</XmlOnly>
  <PrinterDetails>
    <PrinterName />
    <PrintCopies>1</PrintCopies>
    <PrintCollate>True</PrintCollate>
  </PrinterDetails>
  <EmailDetails>
    <EmailFromAddress />
    <EmailToAddress />
    <EmailCCAddress />
    <EmailToAddress />
    <EmailBodyText />
  </EmailDetails>
</DocumentControl>
```

XML In

Valid values for the `DocumentType` element are either `L` (local) or `F` (foreign) and the format code selected must have been created for the specified purchase order document type.

```
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>ONLINE</Function>
    <DocumentType>L</DocumentType>
    <Format>0</Format>
    <Reprint>Y</Reprint>
  </Option>
  <Filter>
    <OrderNumber FilterType="S" FilterValue="458" />
  </Filter>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Document>
    <Status />
    <DocumentGuid/>
    <ErrorMessage/>
    <StatusPreview/>
    <DocumentHex>HEX Encoded PDF file</DocumentHex>
  </Document>
</Query>
```

Quotations

Supported values for the `DocumentType` element are `Quotation - Single`, `Quotation - Multiple line` and `Quotation - Multiple column`.

XML Parameters

```
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl>
  <DocumentType>Quotation - Single</DocumentType>
  <Print>False</Print>
  <Email>False</Email>
  <Preview>True</Preview>
  <XmlOnly>False</XmlOnly>
  <PrinterDetails>
    <PrinterName />
    <PrintCopies>1</PrintCopies>
    <PrintCollate>True</PrintCollate>
  </PrinterDetails>
  <EmailDetails>
    <EmailFromAddress />
    <EmailToAddress />
    <EmailCCAddress />
    <EmailToAddress />
    <EmailBodyText />
  </EmailDetails>
</DocumentControl>
```

XML In

```
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <Function>ONLINE</Function>
    <Format>0</Format>
    <Reprint>Y</Reprint>
    <DocumentDate />
    <PrintDefaultOffer />
    <PrintOffer1>Y</PrintOffer1>
    <PrintOffer2 />
    <PrintOffer3 />
    <PrintOffer4 />
    <PrintOffer5 />
  </Option>
  <Filter>
    <Quotation FilterType="S" FilterValue="951" />
  </Filter>
</Query>
```

XML Out

```
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Document>
    <Status />
    <DocumentGuid/>
    <ErrorMessage/>
    <StatusPreview/>
    <DocumentHex>HEX Encoded PDF file</DocumentHex>
  </Document>
</Query>
```

Factory documentation

XML Parameters

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<DocumentControl>
  <DocumentType>Factory Documentation</DocumentType>
  <Print>False</Print>
  <Email>False</Email>
  <Preview>True</Preview>
  <XmlOnly>False</XmlOnly>
  <PrinterDetails>
    <PrinterName />
    <PrintCopies>1</PrintCopies>
    <PrintCollate>True</PrintCollate>
  </PrinterDetails>
  <EmailDetails>
    <EmailFromAddress />
    <EmailToAddress />
    <EmailCCAddress />
    <EmailToAddress />
    <EmailBodyText />
  </EmailDetails>
</DocumentControl>
```

XML In

```xml
<?xml version="1.0" encoding="Windows-1252"?>
<Query>
  <Option>
    <DocumentNumber>1</DocumentNumber>
    <Format>0</Format>
  </Option>
  <Filter>
    <Job FilterType="S" FilterValue="951" />
  </Filter>
</Query>
```

XML Out

```xml
<?xml version="1.0" encoding="windows-1252" ?>
<Query>
  <Document>
    <Status />
    <DocumentGuid/>
    <ErrorMessage/>
    <StatusPreview/>
    <DocumentHex>HEX Encoded PDF file</DocumentHex>
  </Document>
</Query>
```

Sample code using the sales order document types

The following sample code is provided to assist you in using the API.

The sample code is provided in C# and is specifically for sales order document types. It needs to be adjusted if used for other document types.

Creating the SYSPROSRSClient object

```
// Replace "localhost:20140" with the endpoint of your
// SRS reporting host service
// See Reporting service field on the Reporting tab in System Setup
SYSPROSRSClientLibrary.SYSPROSRSClient _sysproSRSClient =
new SYSPROSRSClientLibrary.SYSPROSRSClient("localhost:20140")
```

Authenticate to the service using operator and company code

The passwords are not validated using this function.

The custom or third party application must handle authentication, should this be required.

```
// The following assumes that you have set up the OperatorCode and
// CompanyCode method arguments with appropriate values.
// Passwords are not required as the host service is a trusted application
string sessionId = "";sessionId = _sysproSRSClient.AuthenticateSYSPROUser(OperatorCode, "",
                                                                           CompanyCode, "");
```

Authenticate to the service using Logon GUID

You would only use this function if you already have a session ID that you want to use.

This method requires operator and company password.

```
// // Create the Load Balancer instance using the e.Net Load Balancer end point
// See Server name and REST port fields on the e.net service details tab
//   in System Setup
// Requires "using SYSPROWCFServicesClientLibrary40;"
SYSPROWCFServicesPrimitiveClient WCFNETTCP;
WCFNETTCP = new SYSPROWCFServicesPrimitiveClient(WCFAddress,
                                                 SYSPROWCFBinding.NetTcp);

// Generate the session id with the Logon function to the relevant
//   SYSPRO instance
string GUID = WCFNETTCP.Logon(OperatorCode, OperatorPassword,
                              CompanyCode, CompanyPassword,
                              "", "", SYSPROInstance, "");

// Validate the session id with the API
string sessionId = "";
sessionId = _sysproSRSClient.AuthenticateSYSPROGuid(GUID);
```

Determine the available document options

```
// Validate the session id with the API
sessionId = _sysproSRSClient.AuthenticateSYSPROGuid(GUID);
StringBuilder xmlIn = new StringBuilder();
xmlIn.Append("<Query>");
xmlIn.Append("<Option>");
xmlIn.Append("<Function>GETDOCDET</Function>");
xmlIn.Append("<IncludeFormatDetails>Y</IncludeFormatDetails>");
xmlIn.Append("</Option>");
xmlIn.Append("<Filter>");
xmlIn.AppendFormat("<OrderNumber FilterType='S' FilterValue='{0}' />", 791);
// Where 791 is the Sales Order number
xmlIn.Append("</Filter>");
xmlIn.Append("</Query>");
string documentOptions = _sysproSRSClient.DetermineDocumentOptions(
sessionId, "SalesOrder", xmlIn.ToString());
// See documentation for sample XML out
```

Produce a document for sales order invoice

```
//Produce a document for sales order invoice
StringBuilder xmlParam = new StringBuilder();
xmlParam.Append("<DocumentControl>");
xmlParam.Append("<DocumentType>SalesOrder</DocumentType>");
xmlParam.Append("<Print>False</Print>");
xmlParam.Append("<Preview>True</Preview>");
xmlParam.Append("<XmlOnly>False</XmlOnly>");
xmlParam.Append("<PrinterDetails>");
xmlParam.Append("<PrinterName />");
xmlParam.Append("<PrintCopies>1</PrintCopies>");
xmlParam.Append("<PrintCollate>True</PrintCollate>");
xmlParam.Append("</PrinterDetails>");
xmlParam.Append("<EmailDetails>");
xmlParam.Append("<EmailFromAddress />");
xmlParam.Append("<EmailToAddress />");
xmlParam.Append("<EmailCcAddress />");
xmlParam.Append("<EmailSubject />");
xmlParam.Append("<EmailBodyText />");
xmlParam.Append("</EmailDetails>");
xmlParam.Append("</DocumentControl>");
StringBuilder xmlIn = new StringBuilder();
xmlIn.Append("<Query>");
xmlIn.Append("<Option>");
// This is always 'ONLINE'
xmlIn.Append("<Function>ONLINE</Function>");
// See Document Type Codes
xmlIn.Append("<DocumentType>I</DocumentType>");
// Format code selected to print with - see output from DetermineDocumentOptions
xmlIn.Append("<Format>0</Format>");
xmlIn.Append("<Reprint>Y</Reprint>");
Print/Reprint flag - see output from DetermineDocumentOptions
xmlIn.Append("</Option>");
xmlIn.Append("<Filter>");
xmlIn.AppendFormat("<OrderNumber FilterType='S' FilterValue='{0}' />",791);
//Where 791 is the Sales Order number
xmlIn.AppendFormat("<InvoiceNumber FilterType='S' FilterValue='{0}' />",100506);
// Where 100506 is the Invoice number
xmlIn.Append("</Filter>");
xmlIn.Append("</Query>");
string xmlOut = _sysproSRSClient.ProduceDocument(
sessionId, xmlParam.ToString(), xmlIn.ToString());
// See documentation for sample XML out
```

Retrieve the document content from the HEX encoded string from the output XML

```
byte[] docByte = GetByteStringFromOutput(xmlOut);
/// <summary>
/// Function that converts a Hex encoded string to an array
/// of unsigned integers that represents the ASCII bytes
/// </summary>
private static byte[] GetByteStringFromOutput(string XMLOut)
{
 byte[] bytes = new byte[] { };
try
{
XDocument XOut = XDocument.Parse(XMLOut);
// Use Linq to get the HEX encoded document string
XElement docHex = XOut.Descendants().Where(n =>
                  n.Name == "DocumentHex").First();
if (docHex == null)
{
return bytes;
}
string HexString = docHex.Value;
// Get the number of characters in the string
int NumberChars = HexString.Length;
// Each byte is derived from 2 characters in the input
// that together represent the HEX value of the byte
bytes = new byte[NumberChars / 2];
for (int i = 0; i < NumberChars; i += 2)
{
// The 16 in the following conversion indicates that this
// is converting from HEX, or Base 16.
bytes[i / 2] = Convert.ToByte(HexString.Substring(i, 2), 16);
}
return bytes;
}
catch { throw; }
}
```

Logoff from the SYSPRO session

```
// If you authenticated using a OperatorCode and
// CompanyCode then you might want to logoff.
_sysproSRSClient.LogoffUserSession(sessionId);
```

Details per document type

Sales Orders

Out Determine

SalesOrder

This is the sales order number.

OrderStatus

This is the sales order status flag.

Sales order status codes

Following the entry of a sales order document into the system, a status code is assigned to the order header indicating its progress within the system.

The status code held against an order determines which functions can be performed at that point, particularly during maintenance and printing.

The current order status is stored in the SorMaster table.

The status may also be stored in the SorMasterRep table, but may not be the same as the status in the SorMaster table.

> **FOR EXAMPLE:**
>
> For an SCT order, the status in the SorMasterRep table is the order status at the time of performing the transfer, whereas the order status in SorMaster is the current status.

The following are the possible statuses that can be assigned to a sales order header:

Status	Description
0 - ORDER IN PROCESS	The entered order is currently being maintained by another user or the **End order** function was not performed (e.g. a power failure occurred during the entry or maintenance of an order). This status can be reset to the previous status using the **Order Maintenance** function.

Status	Description
1 - OPEN ORDER LINE	An order has been entered into the system. A delivery note must be printed before the order can be released for invoicing. Next status: 4 - IN WAREHOUSE The following actions place an order into a status of 4 - IN WAREHOUSE: ▪ Printing a delivery note upon ending the order. ▪ Printing delivery notes using the **Document Print** program for sales orders in 1 - OPEN ORDER LINE. ▪ Using the **Sales Order Entry** program to manually change the order status from 8 - TO INVOICE to 4 - IN WAREHOUSE

Status	Description
2 - Open back order	An order or billing contains lines which could not be shipped owing to a stock shortage. The order or billing may have been partially invoiced, but there are remaining lines on back order. Releases for a **Blanket Sales Order** have been accepted. An order or billing has to pass through the invoicing process at least once to become an open order. You can cancel an order that is in a status 2 - Open back order providing that the order has been partially invoiced and the **Cancel open order previously invoiced** setup option is enabled (*Setup Options* > *Preferences* > *Distribution* > *Sales Orders*). Your configuration of the **Back orders** setup option affects how this status is updated (*Setup Options* > *Preferences* > *Distribution* > *Sales Orders*): ■ If the option is set to **Manual**, then the order is set to a status of 2 - Open back order. ■ If the option is set to **Automatic**, then the order is set to a status of 3 - Released back order. Next status: ■ 3 - Released back order (for orders) ■ 8 - To invoice (for billings)

Status	Description
3 - Released back order	An open back order now has items to be shipped and has been released for delivery note printing. Billings are released into status 8 - To invoice, not status 3 - Released back order. The next delivery note print will produce a delivery note document. Your configuration of the **Back orders** setup option affects how this status is updated (*Setup Options > Preferences > Distribution > Sales Orders*): ■ If the option is set to **Manual**, then the order is set to a status of 2 - Open back order. ■ If the option is set to **Automatic**, then the order is set to a status of 3 - Released back order. Next status: 4 - In warehouse
4 - In warehouse	A delivery note has been printed for an open order or a released back order. Next status: 8 - To invoice
8 - To invoice	A billing has been entered. A back ordered billing has been released. An order in the warehouse has been released. Next status: ■ 9 - Complete (if completely shipped after invoicing) ■ 2 - Open back order (if any back orders remain)
8 - To transfer	A **Supply Chain Transfer** order has been entered. Next status: 9 - Complete

Status	Description
F - Forward order	The order is indefinite and must be released before it is processed as a normal order. Next status: 1 - Open order line When releasing a Forward order, the quantity is placed in back order when any of the following are true: ■ The sales order entry preference **Automatically put order qty into back order** is enabled. ■ There is insufficient stock and negative stock is not allowed for the selected warehouse. ■ Multiple bins are enabled (irrespective of whether stock is being reserved). ■ The item is traceable, ECC-controlled, or serialized. ■ The line is for items that are parents or components of kits (i.e. a single level BOM structure of parent or component exists). Components cannot automatically be released because the stock allocations are required before the lines can be shipped. ■ The Packaging option **Requested and mandatory** or **Requested but optional** are selected. These lines must be manually shipped to ensure that packaging details are entered.
S - In suspense	An order or billing has failed credit checking. An order or billing has been placed on hold. Next status: The order reverts to the status in process (0 - Order in process) prior to the order being suspended.

Status	Description
9 - Complete	A sales order has been fully invoiced and will be removed from file when the **Order Purge** program is run. When a **Blanket Sales Order** is complete in terms of invoicing, it is set to a status of S - In suspense while waiting for the next release. It is not set to a status of 9 - Complete.
* Cancelled	An order was canceled during entry (i.e. before it was completely entered into the system) and will be removed from the file when the **Order Purge** program is run.
\ Cancelled	An order was canceled prior to invoicing and will be removed from the file when the **Order Purge** program is run.
R - Released to picking	Indicates that an order has been released to picking. An open order (status 1 - Open order line) which contains lines for picking, would typically move to a status R - Released to picking and then P - In picking. Sales orders can only be invoiced when the picking cycle has been completed.
P - In picking	Indicates that the order is currently within the picking process. Once picking has completed, the order would return to a status 1 - Open order line, therefore enabling you to add additional lines if required, or proceed to the next required stage.

Translated_OrderStatus

This is the translated description of the order status.

ActiveFlag

This flag determines whether the sales order is active.

- Blank - Yes
- N - No
- Z - To be archived

TranslatedActiveFlag

This is the translated description of the active flag.

CancelledFlag

This flag determines whether the sales order is cancelled.

- Blank - No
- Y - Yes

Translated_CancelledFlag

This is the translated description of the cancelled flag.

SODocumentType

This determines the document type, which can be one of the following:

- B - Billing
- O - Order
- C - Credit note
- D - Debit note

Translated_SODocumentType

This is the translated description of the document type.

CanPrintAcknowledgement

This flag determines whether the order acknowledgment document may be printed.

- True - Yes can be printed
- False - No can't be printed

CanRePrintAcknowledgement

This flag determines whether the order acknowledgment document may be reprinted.

- True - Yes can be reprinted
- False - No can't be reprinted

CanPrintDeliveryNote

This flag determines whether the delivery note document may be printed.

- True - Yes can be printed
- False - No can't be printed

CanRePrintDeliveryNote

This flag determines whether the order acknowledgment document may be reprinted.

- True - Yes can be reprinted
- False - No can't be reprinted

CanGenerateInvoice

This flag determines whether an invoice can be generated from the sales order.

- True - An invoice can be generated from the sales order
- False - An invoice can't be generated from the sales order

CanPrintDispatchNote

This flag determines whether the dispatch note document may be printed.

- True - Yes can be printed
- False - No can't be printed

CanRePrintDispatchNote

This flag determines whether the order acknowledgment document may be reprinted.

- True - Yes can be reprinted
- False - No can't be reprinted

CanGenerateDispatchNote

This flag determines whether a dispatch note can be generated from the sales order.

- True - A dispatch note can be generated from the sales order
- False - A dispatch note can't be generated from the sales order

Documents

This is the list of dispatch notes details for the specified sales order and the list of any detail lines for a given order that has not yet been invoiced.

Dispatch

This lists any dispatch notes that were generated for the order.

DispatchNote

This is the dispatch note number.

DispatchNoteStatus

This is the dispatch note status flag.

Translated_DispatchNoteStatus

This is the translated description of the dispatch note status.

DispatchInvoice

This is the dispatch note invoice.

Invoice

This lists any invoices that have been generated for the sales order.

InvoiceNumber

This is the invoice number.

InvoiceSource

This is source document from which the invoice was generated.

- O = Sales order

Translated_InvoiceSource

This indicates the description of the invoice source.

- O - Order
- M - Multiple dispatch
- C - Consolidate dispatch
- S - SCT
- T - Dispatch SCT

DateLastInvPrt

This is the date when the invoice was last printed.

Formats

This indicates the list of formats that can be used to print a sales order document.

Format

This indicates the format details.

DocumentType

This flag determines the sales order document type.

- Invoice
- Delivery Note

FormatCode

This flag determines the sales order format code.

- a - z
- A - Z

FormatName

This is a description of the format, as supplied by the user.

In Produce

Query

The XML root node. This includes the option and filter to determine how the document should be generated.

Option

These are the options used to determine how the statement is generated.

Filter

This indicates the filter selection. A combination of the filters should equate to a single document.

OrderNumber

This is the sales order number filter.

- Filter type S - single

InvoiceNumber

This is the invoice number used to reprint a single invoice.

DeliveryNote

This is currently not applicable.

DispatchNote

This is the dispatch note number filter.

- Filter type S - single

DocumentDate

This is the date the document was captured.

DocumentType

This is the document type, which can be one of the following:

- D - Delivery note
- I - Invoice
- O - Acknowledgement
- N - Dispatch Note

DocumentFormat

This is the format of the document, which can be one of the following:

- 0 - 9
- A - Z
- a - z

Function

This is the print function:

- ONLINE - synchronously generates a single document

PrintTranslatedText

This translates foreign text and notes for multi-language option.

Reprint

This flag determines whether the invoice may be reprinted.

- Y - reprint

eSignature

This is currently not applicable.

AR Statements

Out Determine DocumentInformation

This indicates the selected format options details and the configuration against the template assigned to the format.

Format

This flag is returned by the `GETFMTS` function.

DocumentType

This is the SRS Document Type.

- Invoice
- get other document types...

FormatCode

This is the Format code that is returned by the `GETFMTDET` function call.

DocumentIndex

This is the index to the document entry in the document print control file that is returned by the `GETFMTDET` function call.

DocumentDescription

This is the document descriptionthat is returned by the `GETFMTDET` function call.

SchemaFile

Name of the schema file used by the document that is returned by the `GETFMTDET` function call.

SchemeServerPath

This is the path to the schema file on the application server that is returned by the `GETFMTDET` function call.

TemplateFile

This is the name of the template file used by the document that is returned by the `GETFMTDET` function call.

TemplateServerPath

This is the path to the template file on the application server that is returned by the `GETFMTDET` function call.

PrintFlags

These are the internal print flags.

CustomSubject

This is the custom email subject line.

CustomPrintFile

This is the custom print file name.

UseSQLDriver)

This lets you select to use the SQL driver instead of the XML driver.

- True - Use SQL driver
- False - Use XML driver

PrinterName

This is the name of the first printer defined against the document.

PrinterCopies

This is the number of copies that will be printed.

PrinterCollate

This lets you collate print copies.

- True - Collate copies
- False - Don't collate copies

FormatName

This is the document format description from the format control record.

In Produce Query

The XML root element. This includes the option and filter to determine how the document should be generated.

Option

This indicates the options used to determine how the statement should be generated

Filter

This indicates the filter selection.

Filters should equate to a single document, and currently there is one filter selection which is a customer.

Customer

This indicates the customer for whom statements are printed.

Function

This Indicates the business object function to be performed.

Currently ONLINE is the only function available.

Format

This is the document format code.

StatementAsOf

This defines the financial period for which the statement will be printed.

- C - Current
- P - Previous period 1
- 2 - Previous period 2

If no value is supplied, a default of 'C - Current' will be used.

StatementDate

This is the ageing date that will be applied to the transactions.

The date will default to the relevant ageing date as per the StatementAsOf option supplied, i.e.

- If either **P - Previous period** or **2 - Previous period 2** is set, the ageing dates for these periods will be applied.
- If this is set to **C - Current period**, the date will default to the company date.
- A manually entered date overwrites any default dates.

StatementAgeing

This indicates what ageing method should be applied when ageing transactions.

- S - Statement
- A - Aged statement
- I - Invoice date
- D - Invoice due date

If no value is supplied, the ageing option defined in the AR setup program will be used.

ConsolidateSub)

This indicates whether to list invoices for attached sub-account customers when printing statements for master account customer.

IncludeAttached

This indicates whether statements are printed for sub-account customers.

If you select **Y – Yes** at **IncludeAttached** and **ConsolidateSub**, the invoices for the attached sub-account customers are printed twice - once on the statement for the master account and once on the statement for the sub-account.

This is assuming that both the master and sub-account customer numbers are included in the range of customers for which statements must be printed.

> If this is set to **Y – Yes**, the **Currency value** in the Statement Information (reflect on the xml out) will not balance back to the values displayed in the **At a Glance - Accounts Receivable**

AsOpenItem

- Y - Yes - prints statement for balance brought forward customer as if it were open-item. i.e. all unpaid invoices are listed on the statement.
- N - No - statements for balance forward customer are printed with a brought forward total followed by only current invoices being listed.

BalanceType

This indicates whether customer with certain balances should be included.

- A - All
- D - Debit
- C - Credit

If no value is supplied then a default value of **A - All** will be used.

MinimumBalance

This indicates the amount which a customer's balance must exceed before a statement is printed.

SalesMessage

A message of up to 50 characters that you want printed on the statement.

Purchase Orders

Out Determine

CanPrintPurchaseOrder

This indicates that the specified purchase order can be printer based on the purchase order status.

- True - the selected purchase order is ready to be printed.

CanRePrintPurchaseOrder

This indicates that the specified purchase order can be re-printed based on whether the purchase order has been printed.

- True - the purchase order has been printed and can now be reprinted.

DocumentControl

The XML root element.

DocumentInformation

This is the XML tag which includes the selected purchase order details including the status of the purchase order and the list of formats that can be used when printing a purchase order.

DocumentType

SRS Document Type, e.g.

- L - Purchase Orders - Local
- F - Purchase Orders - Foreign
- R - Purchase Orders - Requisitions
- C - Blanket Contract

DocumentTypeDescription

This is the description of the SRS Document type.

- Purchase Orders - Local

FormatCode

This is the code for the document format.

FormatName

Document format name.

- P/order - Local

Formats

This indicates the list of the formats that can be used to print a purchase order.

This includes the document type (purchase order foreign or local) and the format codes per document type.

PurchaseOrder

This indicates the purchase order key to be printer.

PurchaseOrderActiveFlag

This indicates whether the purchase order status is active.

PurchaseOrderCancelledFlag

This indicates whether the purchase order status is cancelled.

PurchaseOrderStatus

This indicates the status of the purchase order.

PurchaseOrderType

This indicates the code of the purchase order type.

Translated_OrderStatus

This • indicates the description for the order status.

- 0 - In process
- 1 - Ready to print
- 4 - Order printed
- 9 - Completed
- * - Cancelled

Translated_PurchaseOrderType

This indicates the description of the order type.

- L - Local
- I - Import
- O - Other

In Produce

Function

Print function. 'ONLINE' - synchronously generate a single document.

DocumentType

This indicates the purchase order document type.

- L - Purchase Orders - Local
- F - Purchase Orders - Foreign
- R - Purchase Orders - Requisitions
- C - Blanket Contract

Format

This indicates the purchase order Format code.

- 0 - 9
- A - Z.

This is required for the `GETFMTS` function only.

Reprint

This is reprint flag.

- Y - reprint a quotation

Filter

This indicates the filter selection for the order to be printed.

Filter selection ensures a single purchase order is printed.

OrderNumber

This is the purchase order number filter.

- Filter Type 'S' - single

Quotations

Out Determine

CanPrintQuotation

This indicates that the specified quotation can be printer based on the purchase order status.

- True - the selected quotation is ready to be printed.

CanRePrintQuotation

This indicates that the specified quotation can be re-printed based on whether the quotation has been printed.

- True - the quotation has been printed and can now be reprinted.

DocumentControl

The XML root node.

DocumentInformation

The XML tag which includes the selected quotation details including the status of the quotation and the list of formats that can be used when printing a quotation.

Quotation

This indicates the quotation to be printed.

QuotationVersion

This indicates the version number of the specified quotation.

QuoteStatus

This indicates the status of the specified quotation.

Translated_OrderStatus

This indicates the description of the quotation status.

- 0 - In progress
- 1 - Ready for printing
- 2 - Printed
- 4 - Confirmed
- H - On hold
- R - Rejected
- S - Superseded - new version
- \ - Cancelled

In Produce

DocumentDate

This is the date of the document.

Filter

This indicates the filter selection for the quotation to be printed.

Filter selection ensures a single quotation.

Quotation

This indicates the quotation to be printed.

Format

This is the print function.

- ONLINE - synchronously generate a single document.

PrintDefaultOffer)

This lets you print the default offer.

- D - Print
- N - Do not print

PrintOffer1

This lets you print offer number 1.

- 1 - Print
- N - Do not print

PrintOffer2

This lets you print offer number 2.

- 2 - Print
- N - Do not print

PrintOffer3

This lets you print offer number 3.

- 3 - Print
- N - Do not print

PrintOffer4

This lets you print offer number 4.

- 4 - Print
- N - Do not print

PrintOffer5

This lets you print offer number 5.

- 5 - Print
- N - Do not print

Quotation

This is the quotation number filter type.

- S - single

Reprint

This is the reprint flag.

- Y - reprint a quotation

Factory Documentation

Out Determine

CustomPrintFile

This is the custom print file name.

CustomSubject

This is the custom email subject line.

DocumentControl

The XML root.

DocumentInformation

The XML tag which includes the selected job details including the status of the job and the list of formats that can be used when printing a job.

DocumentDescription

Description against the document in the document print control files that is returned by the `GETFMTDET` function.

DocumentIndex

This is the index to the document record in the document print control files that that is returned by the `GETFMTDET` function.

DocumentType

SRS Document Type.

- Quotation - Single

DocumentNumber

This indicates the document number according to which the job(s) must be printed.

FormatCode

This indicates the document format within the document number according to which the job must be printed.

PrintFlags

These are PDF encryption details that are used by the process generating the PDF to encrypt the document if it is set against the format being used.

PrinterCollate

Print copies collated.

- True
- False

PrinterCopies

This is the number of copies to print.

PrinterName

Name of the first printer defined against the document

SchemaFile

Schema file name - without path - used by the document type that is returned by the `GETFMTDET` function.

SchemaServerPath

Full path to the schema used by the document type that is returned by the `GETFMTDET` function.

TemplateFile

This is the Crystal RPT file name - without path - that used by the document and returned by the `GETFMTDET` function.

TemplateServerPath

This is the full path to the Crystal RPT used by the document and returned by the `GETFMTDET` function.

UseSQLDriver

Use SQL driver instead of XML driver

- True
- False

In Produce

DocumentNumber

This is the document number according to which the job(s) must be printed.

- 1 - 4

Filter

This indicates the job for which the XML/document is generated.

The filter type is always **single** and filter value is **Job**.

Job

This is the job filter type.

- S - single

Format

This is the document format.

- 0 - 9
- A - Z

Option

This indicates the options used to determine how the job should be generated and further filters using the state of the job.

Query

The XML root element.

This includes the option and filter to determine how the document should be generated.

SYSPRO™

www.syspro.com