Contents

Delivery Manager .................................................................7
  Introduction .........................................................................................7
  Output Management System (OMS) ................................................. 7
  The Delivery Manager Connect interfaces ........................................ 7
  Levels of integration ......................................................................... 8
  Supported SAP releases .................................................................... 9

Sending SAP data to StreamServe ..................................................10

Configuring the SAP system for XOM ..............................................13
  Configuring the DM Server and Client integration ............................14
    Configuring a Real Output Management System (ROMS) ...............14
    Configuring a Logical Output Management System (LOMS) ..........15
    Configuring the DM Server destination ...........................................18
  Configuring the DM Command and Client integration ......................21
    Configuring a Real Output Management System (ROMS) ...............21
    Configuring a Logical Output Management System (LOMS) ..........22
      Defining command set for a LOMS for the Command Line and Client .....24
    Delivery Manager command options .............................................26
  Configuring an output device for the Delivery Manager ....................27
    Exporting an OMS definition .........................................................29

Configuring StreamServe for the XOM interface ............................31
  The DM Server ................................................................................32
    Transferring an output request from SAP to an OMS .......................32
    Configuring the DM Server .............................................................33
      Determining the system Id ............................................................38
    Starting the DM Server .................................................................38
    The DM Server logging .................................................................38
    Testing the DM Server connection ................................................39
    Error handling ...............................................................................40
    Starting the DM Sender .................................................................40
      Starting the DM Sender in Windows .............................................40
      Starting the DM Sender in UNIX ................................................41
  The DM Client ..................................................................................43
    Transfer of status information from an OMS to a SAP system ..........43
    Configuring the DM Client ............................................................44
      Determining the system Id ............................................................51
    Starting the DM Client .................................................................51
    DM Client logging .........................................................................52
  Configuring the DM Command - Submit ..........................................53
    Transfer of an output request from SAP to an OMS .......................53
    Error handling ..............................................................................53
Sending notifications to SAP ................................................................. 55
SAP Delivery Manager script functions .................................................. 56
  SAPDMDefaultJobErrorSend............................................................... 59
  SAPDMDefaultJobOkSend ................................................................ 60
  SAPDMDeviceNotificationSend .......................................................... 60
  SAPDMDeviceNotificationSendEx ....................................................... 62
  SAPDMDeviceErrorSend ................................................................. 63
  SAPDMDeviceOkSend ..................................................................... 63
  SAPDMJobErrorSend ..................................................................... 64
  SAPDMJobOkSend ......................................................................... 65
  SAPDMJobNotificationSend ............................................................. 65
  SAPDMJobNotificationSendEx ......................................................... 66
  SAPDMDeviceNotificationSendEx .................................................... 67
Sending file based notifications ............................................................. 68
  Job notification file ...................................................................... 68
  Device notification file ................................................................. 71
Return codes .......................................................................................... 73
  Job status codes ........................................................................... 74
  Device status codes ..................................................................... 76
Configuring the StreamServe Project ..................................................... 77
  Configuring a StreamServe Project ..................................................... 77
Configuring the Runtime ..................................................................... 78
  The job and device status mappings used with Output Center .......... 79
Configuring the Platform ..................................................................... 82
  Input connector ........................................................................... 82
  Output connector ........................................................................ 82
  Enable automatic sending of notifications back to SAP ................. 83
Configuring the Project for file based notifications .............................. 85
  Using the strsdm.fcn function file .................................................. 86
    Configuring a StreamServe Project .................................................. 86
    Adding the strsdm.fcn function file to a Resource set .................... 86
Configuring the Runtime .................................................................... 87
  SAPDMJobNotificationOutEx function call .................................... 88
  SAPDMDeviceNotificationOutEx function call ................................ 88
  The job and device status mappings used with the Output Center ..... 88
Configuring the Platform .................................................................... 91
Testing the Delivery Manager configuration ....................................... 93
Testing the DM Server and Client ......................................................... 94
Testing the DM Command ................................................................ 96
Troubleshooting .................................................................................... 97
  Troubleshooting DM Server and Client .......................................... 98
    Errors when submitting a job ....................................................... 98
    DM Client cannot process notifications and stops ....................... 98
  Troubleshooting DM Command ..................................................... 99
    ROMS not available during SAP configuration ............................. 99
Delivery Manager

This guide describes how to configure StreamServe Connect for SAP - Delivery Manager with your SAP system. Delivery Manager is an add-on module to StreamServer.

This guide only contains information specific to the Delivery Manager Connect solution. For general information about StreamServer, see the standard StreamServe documentation.

Delivery Manager is one of four StreamServe Connect solutions available for use with SAP. For information about the other solutions, see the following documentation:

- StreamServe Connect for SAP - Output+
- StreamServe Connect for SAP - E-docs
- StreamServe Connect for SAP - Business Processes

Installation

For information about how to install the StreamServe Connect solutions, see the StreamServe Connect for SAP - Installation Guide.

Introduction

The Delivery Manager provides an efficient and well-integrated connection to the SAP spooling system through the SAP External Output Management (XOM) interface.

With the addition of the Delivery Manager Connect solution, users can not only see that a job has been sent, but also that it has been successfully printed. Using the Delivery Manager Connect solution, customers have a reliable feedback channel that can return detailed job status information back to the SAP system.

Output Management System (OMS)

Output Management Systems (OMS) complement applications by enabling companies to simultaneously track print jobs on multiple printers in various formats, and guarantee that critical documents are actually printed.

The StreamServe output management system, Output Center, is a complement to StreamServer, and was designed and developed to effectively manage enterprise-wide distributed printing in Enterprise Resource Planning (ERP) environments.
The Output Center monitors the entire print environment, and can be structured according to the unique business needs of an organization. With its easy-to-use interface, the Output Center provides key spooling functions such as managing printer status, job status and notification of job completion. The Output Center supports PCL and PostScript/PDF printers from a variety of manufacturers, using industry standard PJL and IPP.

While job tracking is a key task provided by spool management systems, such as the Output Center, Dazel, Macro4 and IBM InfoPrint Manager, the Delivery Manager is the application that updates the SAP spooler system with the actual status tracked by the OMS system.

**Note:** The Delivery Manager can also be used independently from Output Center to send back job status notifications on any StreamServe job.

### The Delivery Manager Connect interfaces

The Delivery Manager supports the SAP XOM interface, which integrates the SAP spool system with an OMS, for example, StreamServe Output Center.

**Command Line interface**

You can use the Command Line interface to transfer output requests from SAP to the OMS.

**Remote Function Call (RFC) Client interface**

You can use the DM Client interface to enable active reporting notifications back to the SAP system. This avoids polling for status information, which can be very resource demanding.

**Remote Function Call (RFC) Server interface**

You use the DM Server interface to transfer output requests from SAP to the OMS. The DM Server interface enables you to work in a distributed environment. While exchanging job data, you avoid accessing the local file systems which can be protected by firewalls or be running under different operating systems.

The DM Server interface uses the DM Sender to send the actual job data to the OMS.

### Levels of integration

Using the Delivery Manager, you can choose your preferred level of integration:

- Command Line interface together with the DM Client interface, so-called Mixed Mode.
- DM Server and DM Client interface.
Supported SAP releases

The Delivery Manager supports the following XOM integration levels in the SAP releases shown:

- **Command Line interface** — SAP system 3.1 and onwards.
- **DM Client interface** — SAP system 4.0 and onwards.
- **DM Server interface** — SAP system 4.6 C and onwards.
Sending SAP data to StreamServe

This diagram illustrates how output from a SAP system is transferred to StreamServer using the XOM interface and the Delivery Manager applications.

**Note:** The Delivery Manager applications can only return device notifications to a SAP system from a device via a spool management system, such as the StreamServe Output Center, Dazel, Macro 4, or IBM Infoprint Manager.

---

**Figure 1  Sending SAP data to StreamServe using the XOM interface**

**Note:** In this scenario, the data output was configured for StreamServe Connect for SAP - Output+ and E-docs.
Submitting a job via the Delivery Manager

You can use a regular device to print a document, such as an invoice or a report, from the SAP system via the Delivery Manager. This device must be configured for the XOM interface.

The job is submitted by the XOM interface from the SAP spooler to the external OMS for further processing, using one of the following:

- DM Client with the Command Line interface (strsdmsubmit application), also known as Mixed Mode. This is recommended for high-volume processing where SAP and StreamServe can share folders, for example using Samba.
- DM Server interface with the DM Sender (strsdmsender service) in non-direct data mode.

Receiving status messages via the Delivery Manager

A status message can be sent back to the SAP system via the XOM interface using the DM Client interface, which actively returns job and device status information. Different status messages can be sent to the SAP system, depending on the job status at the device level. To follow the status of a submitted job, you can use the SAP Output Controller (transaction code /nsp02).

For example, if the job was successful a notification is returned to the SAP system and the job status is changed from Proc. (processing) to Complete. If an error is reported at the spool job level, an error message is returned and the job status is changed from Proc. to Error.

Note: The Delivery Manager cannot retrieve device status by itself. The status information is provided through a spool management system, such as the StreamServe Output Center, Dazel, Macro4, or IBM Infoprint Manager. However, job status messages can be generated directly by a StreamServer application, through scripting.

The following diagram illustrates the order of requests and data flow when using the RFC Server interface and Output Center.
Figure 2  DM Server/Client data flow in non-direct data mode
Configuring the SAP system for XOM

Recommendations
We recommend you install the Delivery Manager and StreamServer before configuring the SAP system.

Levels of integration
You configure your SAP system to use SAP External Output Management (XOM) interface with an external Output Management System (OMS) for the following levels of integration.


**Note:** You only need to configure your SAP system for the level of integration you want to use, see Determining the Delivery Manager interface on page 14.
Configuring the DM Server and Client integration

The DM Server and Client level of integration incorporates the DM Server and Client interfaces.

**Note:** You only need to configure the DM Server and Client integration, if you are using the DM Server and Client interfaces.

**Required activities**

1. **Configuring a Real Output Management System (ROMS)** on page 14
2. **Configuring a Logical Output Management System (LOMS)** on page 15
3. **Configuring the DM Server destination** on page 18
4. **Configuring an output device for the Delivery Manager** on page 27

**Configuring a Real Output Management System (ROMS)**

A Real Output Management System (ROMS) is a definition that specifies the characteristics of an external Output Management System (OMS). A ROMS definition is needed to integrate the SAP spool system into the external OMS.

To create a Real Output Management System (ROMS)

1. Log on to your SAP system.
2 In the transaction box, enter /nspad. The Spool Administration window opens.

3 Click Extended admin and Output management systems.

4 Click Real output management systems. The List of Real Output Management Systems window opens.

5 Click Change.

6 Click Create.

The Create Real Output Management System window opens.

7 In the Real OMS box, enter a name, for example MY_ROMS.

8 In the Description box, enter a description.

9 Specify the ROMS settings for the RFC interface.

### ROMS settings for the RFC interface

<table>
<thead>
<tr>
<th>Tasking</th>
<th>Select RFC Server.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Job status</strong></td>
<td>The status set for the DM Client. Select <strong>Callback</strong>.</td>
</tr>
<tr>
<td><strong>Device status</strong></td>
<td>The status set for the DM Client. If you want device status information to be reported back, select <strong>Callback</strong>.</td>
</tr>
<tr>
<td><strong>Reconfiguration Request</strong></td>
<td>The interval (in seconds) in which the DM Client checks if any changes have been made on the ROMS. For example, 300 seconds.</td>
</tr>
</tbody>
</table>

10 Click **Save** to save the configuration.

### Configuring a Logical Output Management System (LOMS)

You must define at least one Logical Output Management System (LOMS) that refers to the Real Output Management System (ROMS). More than one LOMS, with different types of devices, can reference the same ROMS.

**Prerequisites**

You must define a ROMS before you can define a LOMS. See *Configuring a Real Output Management System (ROMS)* on page 14.
To create a Logical Output Management System (LOMS)
1. Log on to your SAP system.
2. In the transaction box, enter /nspad. The Spool Administration window opens.
3. Click Extended admin and Output management systems.
4. Click Logical output mgmt systems. The List of Logical Output Management Systems window opens.
5. Click Change.
6. Click Create.

The Create Logical Output Management System window opens.
7. In the Logical OMS box, enter a name, for example MYLOMS.
8. In the Description box, enter a description.
9. Specify the LOMS settings for the RFC interface.

<table>
<thead>
<tr>
<th>LOMS settings for the RFC interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Real OMS</strong></td>
</tr>
</tbody>
</table>
| **Tasking target** | The name of the DM Server destination that you must create (<logical_RFC_destination>). Enter the name for the RFC destination, for example STRSOMS. The RFC destination is created later, see Configuring the DM Server destination on page 18.  
**Note:** The name is case sensitive. |

⚠️ You must enter the name of the DM Server destination in the Tasking target box, for example STRSOMS.  
You can not select the DM Server destination using the browse button. If you use the browse button, the server Id is selected, instead of the DM Server destination, and the LOMS connection will not function correctly.
Configuring the DM Server and Client integration

### Configuring the SAP system for XOM

10. Click **Save** to save the configuration.

#### Spool Admin.: Create Logical Output Management System

![Image of Spool Admin. interface]

| Target for callback | Click the browse button to select the server to be used for callback. This is the server Id (<server_Id>).
| **Note**: To enable load-balancing and fail-over you can configure a virtual spool server that you select from this list. |
|---------------------|-----------------------------------------------------------------------------------------------------|
| Jobs                | Select **Callback**. (Only available if you selected Device status > Callback in the ROMS settings.) |
| Devices             | Select **Callback**.
| Tasking             | Select **Data by file** to use non-direct data mode with the strdmsender service. |

11. Click **Extended Config**.

12. Select the **OMS Configuration** tab, and specify the parameter values:

#### Parameter values

| Send Period | The interval (in seconds) between notification updates by the DM Client. For example, **20**. |
Configuring the DM Server and Client integration
Configuring the SAP system for XOM

13 Click **Save** to save the configuration.

14 You can now configure the output device for the Delivery Manager, see *Configuring an output device for the Delivery Manager* on page 27.

### Configuring the DM Server destination

As communication between the SAP system and the Delivery Manager DM Server uses the RFC protocol, when using the DM Server interface, you need to define a DM Server destination in the SAP system that recognizes the Delivery Manager DM Server. The Delivery Manager DM Server runs as a registered DM Server.

#### To create a DM Server destination

1. Logon to your SAP system release 4.6c (or higher), as a user with administrative permissions.

2. In the transaction box, enter `/nsm59`. The Configuration of RFC Connections view is displayed.

<table>
<thead>
<tr>
<th>Parameter values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of events</strong></td>
<td>The maximum number of Events that should be send back by the DM Client each time. For example 100.</td>
</tr>
<tr>
<td><strong>Interval</strong></td>
<td>The interval (in seconds) if the DM Client failed to connect to the SAP system. For example 300.</td>
</tr>
</tbody>
</table>
3 Select the **TCP/IP Connections** folder, then click **Create**. The RFC Destination view is displayed.

4 Specify the DM Server destination settings.

<table>
<thead>
<tr>
<th>DM Server destination settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RFC Destination</strong></td>
</tr>
<tr>
<td><strong>Connection Type</strong></td>
</tr>
<tr>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>

5 If you want to use Unicode support with the RFC Destination, click the **MDMP & Unicode** tab and select **Unicode**.

6 Click **Save**. The RFC Destination screen shows the new RFC destination.

7 On the **Technical Settings** tab, click **Registered Server Program**.

8 In the **Program ID** field, enter the program Id for this RFC destination, such as **strsoms**. This program Id must be a unique Id for the SAP gateway you are using. It must match the Program Id specified in the DM Server at the RFC Gateway.

9 Click **Save**.
### RFC Destination STRSOMSDOS

<table>
<thead>
<tr>
<th>RFC Destination</th>
<th>STRSOMSDOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Type</td>
<td>TCP/IP Connection</td>
</tr>
</tbody>
</table>

**Description**
- Description 1: Connection to StreamServe RFC Server
- Description 2:
- Description 3:

**Administration**

- **Activation Type**
  - [ ] Start on Application Server
  - [ ] Registered Server Program
  - [ ] Start on Explicit Host
  - [ ] Start on Front-End Work Station

**Registered Server Program**

- Program ID: Stream

**Start Type of External Program**

- [ ] Default Gateway Value
- [ ] Remote Execution
- [ ] Remote Shell
- [ ] Secure Shell

**CFI-O Timeout**

- [ ] Default Gateway Value
- [ ] Specific Timeout: Defined Value in Seconds

**Gateway Options**

- Gateway Host: 
- Gateway service: 

---

**Figure 1** The RFC Destination view
Configuring the DM Command and Client integration

The DM Command and Client level of integration incorporates the Command Line interface and DM Client interface.

**Note:** You only need to configure the DM Command and Client integration, if you are using mixed interfaces — the Command Line interface and DM Client interface.

**Required activities**
- Configuring a Real Output Management System (ROMS) on page 21
- Configuring a Logical Output Management System (LOMS) on page 22
- Configuring an output device for the Delivery Manager on page 27

Configuring a Real Output Management System (ROMS)

A Real Output Management System (ROMS) is a definition that specifies the characteristics of an external Output Management System (OMS). A ROMS definition is needed to integrate the SAP spool system into the external OMS.

**To create a Real Output Management System (ROMS)**

1. Log on to your SAP system.
2. In the transaction box, enter /nspad. The Spool Administration window opens.
3. Click Extended admin and Output management systems.

5. Click Change.

6. Click Create.

    The Create Real Output Management System window opens.

7. In the Real OMS box, enter a name, for example RSTRS.
Configuring the DM Command and Client integration
Configuring the SAP system for XOM

8 In the Description box, enter a description.
9 Specify the ROMS settings for the Command Line interface.

### ROMS settings for the Command Line interface

<table>
<thead>
<tr>
<th>Tasking</th>
<th>Select Command Line.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job status</td>
<td>The status set for the DM Client. Select Callback.</td>
</tr>
<tr>
<td>Device status</td>
<td>The status set for the DM Client. Select Callback.</td>
</tr>
<tr>
<td>Reconfiguration request</td>
<td>The interval (in seconds) in which the DM Client checks if any changes have been made on the ROMS. For example 120.</td>
</tr>
</tbody>
</table>

10 Click **Save** to save the configuration.

### Spool Administration: Create Real Output Management System

#### Real OMS
- Name: ROMS
- Description: My command-client ROMS

#### Output Types
- Default: Auto
- Option: Fax

---

#### Configuring a Logical Output Management System (LOMS)

You must define at least one Logical Output Management System (LOMS) that refers to the Real Output Management System (ROMS). Because each LOMS can have a different set of commands, you can create several LOMSs to use with different printer types.
Prerequisites
You must define a ROMS before you can define a LOMS. See Configuring a Real Output Management System (ROMS) on page 21.

To create a Logical Output Management System (LOMS)
1 Log on to your SAP system.
2 In the transaction box, enter /nspad. The Spool Administration window opens.
3 Click Extended admin and Output management systems.
4 Click Display on the Logical Output Management Systems field. The List of Logical Output Management Systems window opens.
5 Click Change.

6 Click Create.

The Create Logical Output Management System window opens.

7 In the Logical OMS box, enter a name, for example LSTRS.
8 In the Description box, enter a description.
9 Select the ROMS that the LOMS will refer to. You created this ROMS in Configuring a Real Output Management System (ROMS) on page 21.
10 Specify the LOMS settings for the DM Client interface.

<table>
<thead>
<tr>
<th>LOMS settings for the DM Client interface</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Jobs</strong></td>
</tr>
<tr>
<td>Select <strong>Callback</strong>.</td>
</tr>
<tr>
<td><strong>Devices</strong></td>
</tr>
<tr>
<td>Select <strong>Callback</strong>. (Only available if you have selected to use queue query in the ROMS configuration).</td>
</tr>
<tr>
<td><strong>Target for callback</strong></td>
</tr>
<tr>
<td>Browse to and select the server to be used for callback.</td>
</tr>
</tbody>
</table>
When defining a LOMS to be used with the Command Line interface, you must create a command set. See Defining command set for a LOMS for the Command Line and Client on page 24.

Click Save to save the configuration.

Defining command set for a LOMS for the Command Line and Client

If you are defining a LOMS to be used with the Command Line interface, you must create a command set for each operating system the LOMS is running on. When an output request is sent to a LOMS, the commands defined in the command set are used to, for example, submit a job to the Delivery Manager or query the status of the job.

Note: All commands are case sensitive and it is important to enter them exactly as specified above.

Escaping for special characters is handled differently in Windows and UNIX:

Windows — The * character is escaped with the \ character. All \ characters are escaped if they occur before a * character. The $ character is translated into a # character. Arguments with special characters or blanks must be enclosed in double quotes.

UNIX — The /, , ' and $ characters are escaped with the / character. Parameters containing special characters or blanks must be enclosed in double quotes.

To define a command set for a LOMS

1 In your SAP system, select the LOMS for which you want to define a command set.

2 Click Change.

3 Click Commands.

4 Double-click the operating system for which you want to define the OMS. If your operating system is not listed, click Create and enter the operating system name.

5 In the Command path, enter the path of the Delivery Manager command binaries, for example:

Windows: C:\Program Files\StreamServe\Applications\SAP connect\5.5.0\sapdm\bin\
UNIX: /opt/sapconnect/applications/sapconnect/sapdm/
6 On the Submit row, enter submit command with parameters as below.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Submit Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td><code>strsdmsubmit C:\Program Files\StreamServe\Applications\SAP connect\5.5.0\sapdm\bin &amp;EI &amp;EG &amp;S &amp;f &amp;Es &amp;P</code></td>
</tr>
<tr>
<td>UNIX</td>
<td><code>start strsdmsubmit /opt/sapconnect/applications/sapconnect/sapdm &amp;EI &amp;EG &amp;S &amp;f &amp;Es &amp;P</code></td>
</tr>
</tbody>
</table>

You can switch location of the &P and &S parameters if you want the SAP device short name (4 characters), instead of the host printer device name (OMS device) to appear in the spool file name.

You can replace &f with &F to provide the full path to the source file. This makes it ignore the source setting in the configuration, which can be useful in a fail-over scenario where the path to the spool directory may change dynamically.

7 Click **Save** to save the configuration.

8 You can now configure the output device for the Delivery Manager, see *Configuring an output device for the Delivery Manager* on page 27.
## Delivery Manager command options

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;EI</td>
<td>SAP spool Id</td>
<td>Internal spool Id. Required when you use RFC callback. The return parameter for identifying an SAP output request.</td>
</tr>
<tr>
<td>&amp;EG</td>
<td>Reply message group</td>
<td>Reply message group Id. Required to group the returned information.</td>
</tr>
<tr>
<td>&amp;S</td>
<td>Device name</td>
<td>The device name defined in the SAP system.</td>
</tr>
<tr>
<td>&amp;f</td>
<td>Document</td>
<td>File name containing the print data.</td>
</tr>
<tr>
<td>&amp;Es</td>
<td>System Id</td>
<td>Id of the calling SAP system.</td>
</tr>
<tr>
<td>&amp;P</td>
<td>OMS device</td>
<td>The host printer device name</td>
</tr>
</tbody>
</table>
Configuring an output device for the Delivery Manager

You can create a new output device, or modify an existing output device for use with the Delivery Manager.

To configure an output device for the Delivery Manager

1. Log on to your SAP system.
2. In the transaction box, enter /nspad. The Spool Administration view opens.
3. Click Output devices. The List of Output Devices window opens.
4. Click Change.
5. Click Create. The Create Output Device window opens.

You can click Create using template to create an output device based on an existing output device configuration.

6. Specify the output device settings.

<table>
<thead>
<tr>
<th>Output device settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output device</strong></td>
</tr>
<tr>
<td><strong>Short name</strong></td>
</tr>
</tbody>
</table>
7 Select the **Device Attributes** tab, and specify the device settings.

<table>
<thead>
<tr>
<th>Device settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device Type</strong></td>
</tr>
<tr>
<td><strong>Device Class</strong></td>
</tr>
</tbody>
</table>

8 Select the **Access Method** tab, and specify the host spool settings.

<table>
<thead>
<tr>
<th>Host spool settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Spool Access Method</strong></td>
</tr>
<tr>
<td><strong>Note:</strong> If you select an access method that requires a spool server, you must specify the spool server on the Device Attributes tab before you save the device definition.</td>
</tr>
</tbody>
</table>
9 Click Save to save the device definition. The device is now ready to be used with StreamServe and the Delivery Manager.

**Note:** Make sure that the printer has been configured with a device type that suits your StreamServe runtime configuration.

### Exporting an OMS definition

If you want to use a modified version of the Delivery Manager configuration in another SAP system, you must export the OMS definition from the SAP system. The exported configuration can be imported into a production system. The configuration is a transport object and can therefore be transported using the standard SAP transport distribution.

**To export an OMS definition**

1 Log on to the SAP system from which the OMS definition is to be exported.
2 In the transaction box, enter /nse38. The ABAP Editor window opens.
3 In the Program box, enter RSPOXOMS.
4 Click Execute. The Saving and Loading of Definitions window opens.
5 In the Export/import file name box, enter the path to the configuration file that you want to export.
6 Select Server or Frontend computer depending on where you want to export the configuration file.
7 Select Export, Execute export, and Generate log.
8 Enter the LOMS and/or ROMS that you want to export.
9 Click Execute.

Note: The log file is displayed, but there is no indication the OMS definition was successfully exported. Check that the OMS definition exists in the path you specified.
Configuring StreamServe for the XOM interface

The SAP XOM interface allows for different levels of integration. Using the Delivery Manager, you can choose one of the following levels of integration:

- DM Server and DM Client
- Command Line interface and the DM Client (mixed mode)

Required activities
For a pure RFC integration (DM Server and DM Client):
- Configuring the DM Server on page 33
- Configuring the DM Client on page 44

For a mixed Command Line and RFC integration:
- Configuring the DM Client on page 44
- Configuring the DM Command - Submit on page 53
The DM Server

You use the Remote Function Call (RFC) Server interface to transfer output requests from SAP to the Output Management System (OMS). In addition to the DM Client functionality, the DM Server interface enables you to work in a distributed environment. While exchanging job data, you avoid accessing local file systems which can be protected by firewalls or be running under different operating systems.

The DM Server interface uses the DM Sender to transfer the job data from SAP. The DM Sender transfers the job data to a service registered by the Service Request input connector. See Input connector on page 82.

Limitations
The DM Server is only available for SAP release 4.6c and onwards.

Required activities
• Configuring the DM Server on page 33
• Determining the system Id on page 38
• Starting the DM Server on page 38
• Starting the DM Sender on page 40

Transferring an output request from SAP to an OMS

When the SAP spooler submits a job using the DM Server, the same tasks are performed by the server as the strsdsmsubmit application used in the Command Line interface. The difference is that the SAP system uses RFC technology to pass the job data to the external OMS. This allows the OMS handling to be distributed to a separate server which does not need a common file system.

This section describes what happens when you print, for example a Purchase Order, using a device configured for the XOM RFC interface.

You use non-direct data to transfer an output file using the DM Server. An output request is sent from SAP to the DM Server with information about the file name and where to find it in the SAP system.

Transferring an output request from SAP to an OMS in non-direct data mode

1 The output request is sent from the SAP spool system via an RFC connection to the external application.
2 The DM Server receives the job handling request, which contains configuration information.
3 The DM Server calls the DM Sender via a port connection.
4 The DM Sender reads the spool data file.
5 The DM Sender sends the data file to the DM Server.
6 The DM Server writes information about the job into the job notification folder or the runtime database.
7 The DM Server sends the job data into the destination directory specified in the configuration file. The output is ready for further processing, and distribution by the OMS to its final destination, such as printer, fax or email.

**Configuring the DM Server**

You configure the DM Server as a service within the RFC Gateway application.

You can create several DM Server instances that are configured independently, if you for example connect to several SAP systems.

**To configure the DM Server**

1 In Control Center, log in to your site.
2 If you already have created an RFC Gateway, right-click the RFC Gateway and select **Configuration**. Continue to step 10.
   
   **Note:** If the RFC Gateway is already running, you must stop the RFC Gateway before you can configure it.
3 Right-click the application domain and select **New Application**. The New Application dialog box opens.
4 In the **Application type** drop-down list, select **RFC Gateway**.
5 In the **Application name** field, enter a name of the RFC Gateway.
6 Optionally, enter a description in the Application description field
7 Click **OK**. The Configuration dialog opens.
8 Specify **Log level for RFCGateway.log**
   
   1 All error messages  
   2 All error and warning messages  
   3 All error, warning, and information messages  
   4 All error, warning, and extended information messages  
   99 Debug level
   
   **Note:** Higher values do not render more detailed logging. The default is 3.
9 Optionally, specify **RFC Connection pool** size, which is the maximum number of RFC connections that can be pooled simultaneously.
**Note:** Do not specify more connections than you expect to use since a big pool size can slow down processing.

10 Click the **Services** field. The browse button is displayed.

11 Click the browse button. The Service Configuration dialog opens.

12 From the drop-down list, select **DM Server** and click **Add**.

13 Specify the DM Server settings.

### DM Server settings

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th>The name of the DM Server service.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>A description of the DM Server service. (Optional)</td>
</tr>
<tr>
<td><strong>Service name</strong></td>
<td>The service name the received file is submitted to. You must either specify this name or the <strong>Target folder</strong>.</td>
</tr>
<tr>
<td><strong>Target folder</strong></td>
<td>The folder that received files are stored in. Click in the field and the browse button to browse to the folder. You must either specify this folder or the <strong>Service name</strong>.</td>
</tr>
</tbody>
</table>

### SAP Connection properties

| **Program ID** | Program Id which must be unique for the SAP gateway, and must match the Id configured for the RFC destination using the transaction code `/nsm59`. It is recommended to only use capital letters for the value of this parameter. |
| **Host name** | Host name where the SAP gateway is running. Typically a gateway runs on every application server. |
| **Gateway service** | SAP gateway name. The name consists of the gateway name and the system number. The system number must be present in the OS services file, for example in:`<sysdrive>\WINDOWS\system32\drivers\etc` **Note:** Instead of the system number you can use the port number which by default is 3300 for system number 00. |
| **User name** | The user name for logging in to the SAP client. |
| **Password** | The password for logging in to SAP client. |
| **Client number** | The client number to log on to. |
| **Language** | The language to use for the client. |
## DM Server settings

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reconnection interval</strong></td>
<td>The interval (in minutes) which the DM Server waits before attempting to restart after losing connection to the SAP system. The number of times the DM Server attempts to restart is specified by the Reconnection retries value.</td>
</tr>
<tr>
<td><strong>Reconnection retries</strong></td>
<td>The number of times the DM Server attempts to restart after losing connection to the SAP system. The interval which the DM Server waits before attempting to restart is specified by the Reconnection interval value.</td>
</tr>
<tr>
<td><strong>Ignore gateway cancellation</strong></td>
<td>If set to <code>True</code>, the DM Server ignores the cancel signal from the SAP system, and continues to attempt to establish a connection according to the Reconnection interval and Reconnection retries values.</td>
</tr>
</tbody>
</table>

## Direct client connection properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application server host</strong></td>
<td>The SAP system host name.</td>
</tr>
<tr>
<td><strong>System number</strong></td>
<td>The SAP system number.</td>
</tr>
</tbody>
</table>

## Load balanced client connection properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System ID</strong></td>
<td>The Id of the calling SAP system, for example ER2. To find the system Id, see <em>Determining the system Id</em> on page 38.</td>
</tr>
<tr>
<td><strong>Message server host</strong></td>
<td>If you use load balanced DM Servers, specify the host that controls the load balancing.</td>
</tr>
</tbody>
</table>
| **Message service** | Specify only if the Message server does not listen on the standard service `sapms<System Id>`. Or, if this service is not defined in the services file, you have to specify the port directly.  
  The services file is located in `<sysdrive>\WINDOWS\system32\drivers\etc` or a corresponding folder. |
| **Logon group** | Optional - Group name of the application server, default `PUBLIC`.                                    |

## Advanced properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAP Router string</strong></td>
<td>If the connection needs to be made through a firewall using a SAPRouter, specify the SAPRouter parameters in the following format: <code>/H/hostname/S/portnumber/H/</code></td>
</tr>
</tbody>
</table>
## DM Server settings

<table>
<thead>
<tr>
<th><strong>SNC Library path</strong></th>
<th>Full path and name of third-party security library to use for SNC communication (authentication, encryption and signatures). Alternatively you can set the environment variable SNC_LIB.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNC ID</strong></td>
<td>Token/identifier representing the external RFC program.</td>
</tr>
<tr>
<td><strong>SNC Partner ID</strong></td>
<td>Token/identifier representing the back-end system.</td>
</tr>
</tbody>
</table>
| **SNC QOP**          | Use one of the following values:  
  1 – Digital signature  
  2 – Digital signature and encryption  
  3 – Digital signature, encryption and user authentication  
  8 – Default value defined by back-end system  
  9 – Maximum value that the current security product supports |
| **Trace level**      | Use one of the following values:  
  0 – off  
  1 – brief  
  2 – verbose  
  3 – full |
| **Partner character size** | Partner character size.  
  The use cases for this option are not very frequent.  
  During the initial handshake, the RFC library obtains the correct value from the back-end and uses it from then on.  
  One rare use case is as follows: The back-end is Unicode and you want to use a non-ISO-Latin-1 user name or password for the initial logon. As the initial handshake is done with ISO-Latin-1, the characters in User name and Password could not be read and logon is denied. In that case, set this option to 2 and the RFC library uses Unicode for the initial handshake. |
### DM Server settings

| Codepage | Similar to Partner character size above.  
|-----------|------------------------------------------------|
|           | Only needed it if you want to connect to a non-Unicode back-end using a non-ISO-Latin-1 user name or password.  
|           | The RFC library uses the codepage for the initial handshake, thus preserving the characters in username/password.  
|           | A few common values are:  
|           | - 1401: ISO-Latin-2  
|           | - 1500: ISO-Latin-5/Cyrillic  
|           | - 1600: ISO-Latin-3  
|           | - 1610: ISO-Latin-9/Turkish  
|           | - 1700: ISO-Latin-7/Greek  
|           | - 1800: ISO-Latin-8/Hebrew  
|           | - 1900: ISO-Latin-4/Lithuanian/Latvian  
|           | - 8000: Japanese  
|           | - 8300: Traditional Chinese  
|           | - 8400: Simplified Chinese  
|           | - 8500: Korean  
|           | - 8600: Thai  
|           | - 8700: ISO-Latin-6/Arabic  
| Note: | The values can be customized in the back-end system. Contact your back-end system administrator before modifying these settings.  
| No compression | By default the RFC protocol compresses tables, when they reach a size of 8 KB or more.  
| | In very special situations, it may be useful to turn this off. For example, if you are transporting very large integer/binary tables with "random" data, where compression would have no effect, while consuming CPU resources.  
| Backup connection properties | You can configure a backup SAP connection in case the standard one fails. The exact same properties are configurable as for the SAP Connection properties.  
| DM Sender connection properties | |
Determining the system Id

In the DM Server settings, you must specify the SAP system name to identify the source and target SAP systems. You can determine the SAP system name from within your SAP system.

To determine the SAP system name

In your SAP system, select System > Status. The SAP system name is displayed in the Database data area.

Starting the DM Server

When you have completed the configuration of the DM Server, you can start the DM Server.

<table>
<thead>
<tr>
<th>DM Server settings</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host name</strong></td>
<td>The host name or IP address where the DM Sender is running.</td>
</tr>
<tr>
<td><strong>Port number</strong></td>
<td>The port through which the DM Server sends the request to the DM Sender in non-data direct mode, and the port the DM Sender listens to for requests from the DM Server. You must use a port number that is free. Since the DM Sender should not be run as a privileged process, you should use a port number greater than 1023.</td>
</tr>
<tr>
<td><strong>Reconnection retries</strong></td>
<td>The number of times the DM Server tries to establish a connection to the DM Sender until final failure.</td>
</tr>
<tr>
<td><strong>Communication time-out</strong></td>
<td>The maximum time in milliseconds the DM server uses in one attempt to establish a connection to the DM Sender.</td>
</tr>
<tr>
<td><strong>User exit scripts</strong></td>
<td>Optionally, browse to a script file. For example, that can trigger an external workflow on an “At Job Receipt” event, where the notification parameters are supplied as arguments to the script execution.</td>
</tr>
<tr>
<td><strong>Script for job data event</strong></td>
<td>Optionally, browse to a script file. For example, that can trigger an external workflow on an “At Job Data Receipt” event, where the notification parameters are supplied as arguments to the script execution.</td>
</tr>
</tbody>
</table>
To start the DM Server from the Control Center
1 Start the Control Center.
2 Right-click the RFC Gateway and select **Start**.
   The RFC Gateway should look as follows, which indicates that it is started.
   If not, view the RFC Gateway log file for more information.

   ![A running RFC Gateway](image)

To start the DM Server in UNIX
1 Use Command Line Utilities to start the RFC Gateway service. See the **Command Line Utilities** documentation.
2 Verify that the DM Server has started, see *The DM Server logging* on page 39.
3 Test the DM Server connection, see *Testing the DM Server connection* on page 39.

The DM Server logging
After starting the RFC Gateway, the platform log and the RFC Gateway log are scanned. If you run Control Center, the result is shown in the output view on separate tabs.

Before testing the DM Server connection, check that the RFC Gateway is running (State: Running and a green arrow icon). If it is not running, view the log file and correct any problems before you attempt to restart the RFC Gateway.

To clear the log window
In the Control Center browser, right-click the RFC Gateway and select **Clear Log**.
The current log tab is cleared. New logging information will be continuously shown.

Testing the DM Server connection
Before sending any data from the SAP system to the DM Server, you should test the DM Server connection with the SAP system.

To test the DM Server connection to the SAP system
1 In the Control Center, start the RFC Gateway. See *Starting the DM Server* on page 38.
2 In the SAP system, use transaction code `/nsm59` to locate the RFC destination you have created.
3 Click **Test Connection** to verify the connection. If no error messages occur, the configuration is correctly defined.

**Note:** If you have a Unicode enabled SAP system, you can also test the Unicode connection.

### Error handling

The following error handling is performed during the submit phase:

1. The DM Server checks whether the correct parameters are passed correctly from the SAP system or not. If not, an error is returned to the SAP system.

2. Before the job data is written to the destination, the DM Server checks if the destination exists. If not, an error is returned to the SAP system.

3. If an error occurs while the DM Server writes the job data into the destination directory at the OMS, an error is returned to the SAP system.

### Starting the DM Sender

The DM Sender service runs local to the SAP spool processor. When it receives a request from the DM Server, the DM Sender checks the **System ID** parameter in the RFC Gateway to determine where to find the SAP spool job file and sends it to the DM Server.

Since the DM Sender resides on another physical server than the DM Server, you must register and start the DM Sender manually.

### Starting the DM Sender in Windows

You must create a DM Sender service that can be scheduled to be started and stopped from the Windows Services manager.

**Prerequisites**

Microsoft .NET Framework 2.0 must be installed on the SAP machine where you register and run the DM Sender.

**To register the DM Sender**

1. Open a command prompt and browse to the Delivery Manager \bin folder. For example, `C:\Program Files\StreamServe\Applications\SAP connect\5.5.0\sapdm\bin`

2. Enter the following command:
   
   ```
   strsdmsenderregtool -reg <senderid> <serverid> <wd> <path_to_exe>
   ```
   
   where:

   ```
   <senderid>  The DM Sender Id. You can use any integer.
   ```
The default display name of the service is:
StreamServe DM Sender <serverid>

For example:
StreamServe DM Sender 1

You can check that the DM Sender is registered in Control Panel > Administrative Tools > Services.

To start the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -start <senderid>

To stop the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -stop <senderid>

To unregister the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -unreg <senderid>

Starting the DM Sender in UNIX
Open a command prompt and enter the following:
> ./start [-background] strsdmsender -wd <wd> -serverid <server_id>

where:

<serverid>
The Server Id of the DM Server that the DM Sender connects to. This specifies the section to be read in the strsdmstart.cfg file.
The number must correspond to an entry in the strsdmstart.cfg file, such as:
//Number identifying each strsdmsender
[serverparameter1]
That means, for serverparameter1, the server_id is 1. For serverparameter2, the server_id is 2, etc.

<wd>
The Delivery Manager working directory with the strsdmstart.cfg file. This is located on the physical server hosting the DM Sender.

<path_to_exe>
The absolute path to strsdmsender.exe

The default display name of the service is:
StreamServe DM Sender <serverid>

For example:
StreamServe DM Sender 1

You can check that the DM Sender is registered in Control Panel > Administrative Tools > Services.

To start the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -start <senderid>

To stop the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -stop <senderid>

To unregister the DM Sender
Open a command prompt and enter the following:
strsdmsenderregtool -unreg <senderid>

Starting the DM Sender in UNIX
Open a command prompt and enter the following:
> ./start [-background] strsdmsender -wd <wd> -serverid <server_id>

where:

<wd>
The Delivery Manager working directory with the strsdmstart.cfg file. This is located on the physical server hosting the DM Sender.
The numeric Server Id of the DM Server that the DM Sender connects to. The number must correspond to an entry in the `strsdmstart.cfg` file, such as:

```
//Number identifying each strsdmsender
[serverparameter1]
```

That means, for `serverparameter1`, `server_id` is 1. For `serverparameter2`, `server_id` is 2, etc.

**Note:** The `strsdm_logmessages.txt` file must be located in the same folder as the `strsdmsender` application.
The DM Client

You use the Remote Function Call (RFC) Client interface to enable active reporting of events back to the SAP system. This avoids polling for recurrent status information, which can be very resource demanding.

The DM Client interface enables job and device status information to be sent back to the SAP system. You can display information about jobs in your SAP system using the SAP Output Controller, transaction code /nsp02.

The DM Client actively reports status to the SAP system over the RFC interface. The DM Client collects several events and sends them to the SAP system at a single callback in order to reduce the load on the system.

DM Client actions

The DM Client performs a sequence of actions to report status back to the SAP system:

1. Establishes a connection to the SAP system.
2. Reads the job notification file from the notification table in the runtime database or from the job notification folder.
3. Updates the status in the SAP system.
4. Removes the job notification file from the notification table in the runtime database or from the job notification folder.
5. Reads the device notification from the notification table in the runtime database or the device notification folder.
6. Updates the status in the SAP system.
7. Removes the device notification file from the notification table in the runtime database or device notification folder.
8. Updates the notification table in the runtime database or the device notification folder with the reported state.
9. The DM Client repeats steps 1 to 8 until a stop signal is received.
10. Closes the connection.

Required activities

- Transfer of status information from an OMS to a SAP system on page 43
- Configuring the DM Client on page 44
- Starting the DM Client on page 51

Transfer of status information from an OMS to a SAP system

The DM Client actively sends status information to the SAP system.
This section describes what happens when the DM Client service sends status information to the SAP system.

### An external OMS transfers status information to an SAP system

1. The DM Client reads the XML created according to what you have specified on the RFC Gateway DM Client service to determine where information about the job and the device could be found.

   See *Transfer of status information from an OMS to a SAP system* on page 43.

2. The DM Client connects to the SAP system using connection parameters you have specified on the RFC Gateway DM Client service.

3. The RFC Client checks the notification table in the runtime database, or the job notification folder, and if the job exists in the database.

   **Note:** The SAP spool Id is the key that links the outgoing job with the correct status information.

   **Note:** The Delivery Manager cannot retrieve job or device status by itself. The status information is provided through a spool management system, such as the StreamServe Output Center. It is also possible to create notifications through StreamServe scripting.

4. The notification is removed from the runtime database, or the job notification folder.

5. The DM Client passes back the status to the SAP system. When you use SAP Output Controller (transaction code `/nsp02`), the status information shown has been sent from the external OMS to the SAP system.

6. The DM Client checks the notification table in the runtime database or in the device notification folder, and if the device exists in the database.

   **Note:** The SAP spool Id is the key that links the outgoing job with the correct status information.

   **Note:** The Delivery Manager cannot retrieve job or device status by itself. The status information is provided through a spool management system, such as the StreamServe Output Center. It is also possible to create notifications through StreamServe scripting.

7. The device status is reported back to the SAP system.

### Configuring the DM Client

You configure the DM Client as a service within the RFC Gateway application.
To configure the DM Client

1. In Control Center, log in to your site.
2. If you already have created an RFC Gateway, right-click the RFC Gateway and select Configuration. Continue to step 10.

   Note: If the RFC Gateway is already running, you must stop the RFC Gateway before you can configure it.

3. Right-click the application domain and select New Application. The New Application dialog box opens.
4. In the Application type drop-down list, select RFC Gateway.
5. In the Application name field, enter a name of the RFC Gateway.
6. Optionally, enter a description in the Application description field.
7. Click OK. The Configuration dialog opens.
8. Specify Log level for RFCGateway.log

   1. All error messages
   2. All error and warning messages
   3. All error, warning, and information messages
   99 Debug level

   Note: Higher values do not render more detailed logging. The default is 3.

9. Optionally, specify RFC Connection pool size, which is the maximum number of RFC connections that can be used simultaneously.

   Note: Do not specify more connections than you expect to use since a big pool size can slow down processing.
10. Click the Services field. The browse button is displayed.
11. Click the browse button. The Service Configuration dialog opens.
12. From the drop-down list, select DM Client and click Add.
13. Specify the DM Client settings:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>DM Client</td>
<td></td>
</tr>
</tbody>
</table>
### The DM Client

#### Configuring StreamServe for the XOM interface

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>A name of this DM Client instance.</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>An optional description of this DM Client instance.</td>
<td></td>
</tr>
<tr>
<td>ROMS name</td>
<td>The name of the Real OMS you have configured in the SAP system.</td>
<td></td>
</tr>
</tbody>
</table>

#### SAP Connection properties

<table>
<thead>
<tr>
<th>User name</th>
<th>The SAP system user name.</th>
<th>dos01</th>
</tr>
</thead>
<tbody>
<tr>
<td>Password</td>
<td>The SAP system user password. Both uppercase and lowercase letters can be used.</td>
<td></td>
</tr>
<tr>
<td>Client number</td>
<td>The SAP Client number.</td>
<td>100</td>
</tr>
<tr>
<td>Language</td>
<td>Language used when communicating with SAP.</td>
<td>EN</td>
</tr>
<tr>
<td>Reconnection interval</td>
<td>Number of seconds the DM Client waits between reconnection retries. See Reconnection retries below.</td>
<td>10</td>
</tr>
<tr>
<td>Reconnection retries</td>
<td>Number of times the DM Client tries to reconnect to SAP after a connection failure.</td>
<td>30</td>
</tr>
</tbody>
</table>

#### SAP Connection type setting

**Select SAP Connection type**
Depending on the selection here, a separate set of options is displayed below:

#### Direct client connection properties

<table>
<thead>
<tr>
<th>Application server host</th>
<th>The name of the SAP server.</th>
<th>sapservnt02</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Number</td>
<td>The system number</td>
<td></td>
</tr>
</tbody>
</table>

#### Load balanced connection properties

| System ID | The system the file is copied to. To find the system Id, see Determining the system Id on page 51. | ER2           |
Configuring StreamServe for the XOM interface

**Message server host**

Load distribution is handled by the message server (there is one message server in an SAP System). When a user logs on, the message server assigns him or her to the SAP application server that currently has the smallest load. This procedure is also used for incoming HTTP requests.

**Message service**

Specify only if the Message server does not listen on the standard service `sapms<System Id>`. Or, if this service is not defined in the services file and you have to specify the port directly.

The services file is located in `<sysdrive>\WINDOWS\system32\drivers\etc` or a corresponding folder.

**Logon group**

An optional name of the SAP application server group. By default it is `PUBLIC`.

**Advanced properties**

**SAP Router string**

If the connection needs to be made through a firewall using a SAPRouter, specify the SAPRouter parameters in the following format: `/H/hostname/S/portnumber/H/`

**SAP Library path**

Full path and name of the third-party security library to use for SNC communication (authentication, encryption and signatures).

Alternatively you can set the environment variable `SNC_LIB`.

**SNC ID**

Token/identifier representing the external RFC program.
### SNC Partner ID

Token/identifier representing the back-end system.

### SNC QOP

Use one of the following values:

- 1 – Digital signature
- 2 – Digital signature and encryption
- 3 – Digital signature, encryption and user authentication
- 8 – Default value defined by back-end system
- 9 – Maximum value that the current security product supports

### Trace level

A value from 0 to 4, where 4 gives the most detailed information. For production, use log level 0.

### Partner character size

Partner character size. The use cases for this option are not very frequent.

During the initial handshake, the RFC library obtains the correct value from the back-end and uses it from then on.

One rare use case is as follows: The back-end is Unicode and you want to use a non-ISO-Latin-1 user name or password for the initial logon. As the initial handshake is done with ISO-Latin-1, the characters in **User name** and **Password** could not be read and logon is denied. In that case, set this option to 2 and the RFC library uses Unicode for the initial handshake.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Example value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codepage</td>
<td>Similar to Partner character size above. Only needed it if you want to connect to a non-Unicode back-end using a non-ISO-Latin-1 user name or password. The RFC library uses the codepage for the initial handshake, preserving the characters in username/password. A few common values are: • 1401: ISO-Latin-2 • 1500: ISO-Latin-5/Cyrillic • 1600: ISO-Latin-3 • 1610: ISO-Latin-9/Turkish • 1700: ISO-Latin-7/Greek • 1800: ISO-Latin-8/Hebrew • 1900: ISO-Latin-4/Lithuanian/Latvian • 8000: Japanese • 8300: Traditional Chinese • 8400: Simplified Chinese • 8500: Korean • 8600: Thai • 8700: ISO-Latin-6/Arabic Note: The values can be customized in the back-end system. Contact your back-end system administrator before modifying these settings.</td>
<td></td>
</tr>
</tbody>
</table>
### No compression

By default, the RFC protocol compresses tables when they reach a size of 8 KB or more. In very special situations it may be useful to turn this off. For example, if you are transporting very large integer/binary tables with "random" data, where compression would have no effect, while consuming CPU resources.

### Host name

Optionally, specify the SAP Gateway host name.

### Backup connection properties

You can configure a backup DM Client in case the standard one fails, with the exact same properties as for the SAP Connection properties.

### Job notifications folder

Path to the notification folder where job status messages are sent.

*Note:* By default, job status messages are stored in the job status schema in the runtime database.

### Device notifications folder

Path to the notification folder where device status messages are sent.

*Note:* By default, device status messages are stored in the device status schema in the runtime database.

### Polling interval

The interval in seconds between scans of the status schemas and, if you use them, the notification folders.

*Note:* This value must not be higher than the Send Period parameter of the LOMS.
Determining the system Id

In the DM Client settings, you must specify the SAP system name to identify the source and target SAP systems. You can determine the SAP system name from within your SAP system.

To determine the SAP system name

In your SAP system, select **System > Status.** The SAP system name is displayed in the Database data area.

Starting the DM Client

If you have configured a DM Client in the RFC Gateway, the DM Client is started when you start the RFC Gateway. If you do not have access to Control Center, you can start the RFC Gateway using the Command Line Utilities.
For more information, see the *Command Line Utilities* documentation.

**To start the RFC Gateway in Control Center**
In Control Center, right-click the RFC Gateway and select *Start*.

**DM Client logging**

After the DM Client has started, logging is registered in `RFCGateway.log` and the `platform.log` file whose content is displayed on separate tabs in Control Center.

**Log level**
The details shown in the log window vary according to the log level specified for the RFC Gateway. For example:

- The highest level is 4, giving the most detailed log information and is generally used during testing.
- The lowest level is 1, and is suitable for production.

**To clear the log window**

- In the Control Center browser, right-click the RFC Gateway and select *Log > Clear log*. The current log file in the Log tab is cleared. New logging information is continuously shown.
Configuring the DM Command - Submit

You can use the Command Line interface to transfer output requests from SAP to the OMS.

Note: To get notifications back about job and device status, you must use the DM Client.

Transfer of an output request from SAP to an OMS

You use the Command Line interface with the `strsdmsubmit` application to transfer output requests from the SAP spool system to the OMS via the XOM interface.

This section describes what happens when you print, for example, a Purchase order, using a device configured for the XOM command line interface.

Transfer of an output request from a SAP system to an OMS

1. The output request is passed from the SAP spool system via a print command.
2. The print command launches the `strsdmsubmit` application.
3. The `strsdmsubmit` application reads the `strsdmstart.cfg` file. The settings you have made in this file specify the values of the source and target parameters for the output. The `strsdmstart.cfg` file is by default located in the following folder:
   `<StreamServe installation>\Applications\SAP connect \<version>\sapdm\bin`
   
   Note: Instead of specifying a target folder in the `strsdmstart.cfg` file, you can specify a command for an alternative way of transferring the output, e.g. over HTTP.
4. The Delivery Manager copies the file from the source directory on the SAP system to the destination directory on the OMS.
5. The output is now available on the destination and is ready for further processing and distribution by the OMS to its final destination (for example printer, fax or email).

Error handling

The following error handling is performed during the submit phase:

1. The `strsdmsubmit` application checks if the correct parameters are passed correctly from SAP, if not an error is returned to the SAP system.
2. Before the job is passed on to the destination, the `strsdmsubmit` application checks if the destination exists and if copying the file was successful. If not, an error is returned to the SAP system.
Configuring StreamServe for the XOM interface
Sending notifications to SAP

The DM Client delivers status information as events concerning output jobs or devices from the external system to the SAP system. The recommended way to let the DM Client retrieve status information is to use the built-in StreamServe script functions that store notifications in the runtime database (e.g. StrsData).

When using the DM Server/Client integration and the Service Request connector, you can also enable automatic XOM notifications, which negates the need for scripting to store the job status notifications in the runtime repository.

See Configuring the StreamServe Project on page 77.

For available script functions, see SAP Delivery Manager script functions on page 56.
SAP Delivery Manager script functions

The SAP Delivery Manager functions are used to send notifications on job status and device status to the runtime repository. The Delivery Manager Client will then asynchronously read the status information and pass it on to the SAP application server.

The table below contains a short description of each script function. For information about syntax, examples, etc., see the full description of each script function.

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAPDMDefaultJobErrorSend</strong></td>
<td>Stores a SAP spool job notification in the runtime repository. The notification states that the job was erroneous. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. <strong>Note:</strong> This function requires that you use the Service Request connector to automatically retrieve the RMG Id, Spool Id, and Device Id.</td>
</tr>
<tr>
<td><strong>SAPDMDefaultJobOkSend</strong></td>
<td>Stores a SAP spool job notification for the currently processed spool job id in the runtime repository. The notification states that the job was successfully printed. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. <strong>Note:</strong> This function requires that you use the Service Request connector to automatically retrieve the RMG Id, Spool Id, and Device Id.</td>
</tr>
</tbody>
</table>
Sending notifications to SAP

<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAPDMDeviceNotificationSend</td>
<td>Stores a SAP spool job notification for the specified device in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. This function allows you to specify detailed notification information. However, in most cases the SAPDMDeviceOkSend and SAPDMDeviceErrorSend functions are sufficient.</td>
</tr>
<tr>
<td>SAPDMDeviceNotificationSendEx</td>
<td>Stores a SAP spool job notification for the specified device in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. This function allow you to specify an Output Center device status code as an input parameter, which the function translates to actual SAP device status parameters.</td>
</tr>
<tr>
<td>SAPDMDeviceErrorSend</td>
<td>Stores a SAP spool job notification for the specified device in the runtime repository. The notification states that the device has an error. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. <strong>Note:</strong> If you need to include detailed information in the notifications you can use the SAPDMDeviceNotificationSend.</td>
</tr>
</tbody>
</table>
### Function Name: SAPDMDeviceOkSend

Stores a SAP spool job notification for the specified device in the runtime repository. The notification states that the device is OK. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

**Note:** If you need to include detailed information in the notifications you can use the `SAPDMDeviceNotificationSend` function.

### Function Name: SAPDMJobErrorSend

Stores a SAP spool job notification for the specified spool job id in the runtime repository. The notification states that the job was erroneous. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

**Note:** If you need to include detailed information in the notifications you can use the `SAPDMJobNotificationSend` function.

### Function Name: SAPDMJobNotificationSend

Stores a SAP spool job notification for the specified spool job id in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

**Note:** This function allows you to specify detailed notification information. However, in most cases the `SAPDMJobOkSend` and `SAPDMJobErrorSend` functions can be sufficient to use.
<table>
<thead>
<tr>
<th>Function name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAPDMDefaultJobErrorSend</strong></td>
<td>Stores a SAP spool job notification in the runtime repository. The notification states that the job was erroneous. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. This function allow you to specify an Output Center device status code as an input parameter, which the function translates to actual SAP parameters.</td>
</tr>
<tr>
<td><strong>SAPDMJobNotificationSendEx</strong></td>
<td>Stores a SAP spool job notification for the specified spool job id in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. This function allow you to specify an Output Center device status code as an input parameter, which the function translates to actual SAP parameters.</td>
</tr>
<tr>
<td><strong>SAPDMJobOkSend</strong></td>
<td>Stores a SAP spool job notification in the runtime repository. The notification states that the job was successfully printed. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server. The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client. <strong>Note:</strong> If you need to include detailed information in the notifications you can use the <code>SAPDMDeviceNotificationSend</code>.</td>
</tr>
</tbody>
</table>

**SAPDMDefaultJobErrorSend**

**Syntax**

```javascript
SAPDMDefaultJobErrorSend();
```

**Description**

Stores a SAP spool job notification in the runtime repository. The notification states that the job was erroneous. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

**Note:** This function requires that you use the Service Request connector to automatically retrieve the RMG Id, Spool Id, and Device Id.

If you use a Service Request connector you can retrieve the values from the `GetConnectorValue()` function:

```javascript
$spoolid = GetConnectorValue("S_SPOOLID");
```
SAP Delivery Manager script functions

Sending notifications to SAP

$rmgid = GetConnectorValue("RMG_ID");
$device = GetConnectorValue("OSPRT");

Returns
N/A

Example
$notification = SAPDMDefaultJobErrorSend();

SAPDMDefaultJobOkSend

Syntax
SAPDMDefaultJobOkSend();

Description
Stores a SAP spool job notification for the currently processed spool job id in the runtime repository. The notification states that the job was successfully printed. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

Note: This function requires that you use the Service Request connector to automatically retrieve the RMG Id, Spool Id, and Device Id.

If you use a Directory connector you can retrieve the values from the GetConnectorValue() function:

$rspoolid = GetConnectorValue("_SPOOLID");
$rmgid = GetConnectorValue("RMG_ID");
$device = GetConnectorValue("OSPRT");

Returns
N/A

Example
$notification = SAPDMDefaultJobOkSend();

SAPDMDeviceNotificationSend

Syntax
SAPDMDeviceNotificationSend(rmgid, device, reportlevel, classcode, unctime, qenabled, penabled, alarm, busy, incomplete, language, message);

| rmgid         | The Reply Message Group Id, for example PR31_01sapserv. |
**Description**

Stores a SAP spool job notification for the specified device in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

| **device** | The device configured in SAP, i.e. the host name, for example DMPRINT. |
| **reportlevel** | To have an influence on performance you can control at which level notifications should be sent.  
01 - Completion  
02 - Problem with intervention  
03 - Problem without intervention  
04 - Status change  
05 - Information (no error)  
09 - All available information  
**Note:** A higher number includes the information sent in lower numbers. |
| **classcode** | The class code.  
01 - Error  
02 - Problem requiring intervention  
03 - Problem not requiring intervention  
04 - Information (no error) |
| **unctime** | UNC date and time stamp in the following format: YYYYMMDDHHMMSS.  
For example:  
20100228124500 |
| **qenabled** | Yes / No.  
Print queue is enabled and not paused. |
| **penabled** | Yes / No.  
Printing is enabled. At least one of the printers can print. |
| **alarm** | Yes / No. |
| **busy** | Yes / No. |
| **incomplete** | Yes / No. |
| **language** | For example EN. |
| **message** | Detailed uninterpreted info. However logged by the SAP system. |
This function allows you to specify detailed notification information. However, in most cases the `SAPDMDeviceOkSend` and `SAPDMDeviceErrorSend` functions are sufficient.

**Returns**
N/A

**Example**
```php
$notification = SAPDMDeviceNotificationSend($rmgid, $sapdevice, "04", "04", ",", ",", ",", ",", ",", ",", ",", "EN", "Ready");
```

**SAPDMDeviceNotificationSendEx**

**Syntax**
```
SAPDMDeviceNotificationSendEx();
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rmgid</code></td>
<td>The Reply Message Group Id, for example PR31_01sapserv.</td>
</tr>
<tr>
<td><code>device</code></td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
<tr>
<td><code>occode</code></td>
<td>An Output Center status code. The function translates this code to valid SAP status codes. See the Delivery Manager documentation for available Output Center status codes.</td>
</tr>
</tbody>
</table>

**Description**
Stores a SAP spool job notification for the specified device in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

This function allow you to specify an Output Center device status code as an input parameter, which the function translates to actual SAP device status parameters.

**Returns**
N/A

**Example**
```php
$notification = SAPDMDeviceNotificationSendEx($rmgid, $sapdevice, "1");
```
SAPDMDeviceErrorSend

Syntax

SAPDMDeviceErrorSend(rmgid, device);

<table>
<thead>
<tr>
<th>rmgid</th>
<th>The Reply Message Group Id, for example PR31_01sapserv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
</tbody>
</table>

Description

Stores a SAP spool job notification for the specified device in the runtime repository. The notification states that the device has an error. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

Note: If you need to include detailed information in the notifications you can use the SAPDMDeviceNotificationSend.

Returns

N/A

Example

$notification = SAPDMDeviceErrorSend($rmgid, $sapdevice);

SAPDMDeviceOkSend

Syntax

SAPDMDeviceOkSend(rmgid, device);

<table>
<thead>
<tr>
<th>rmgid</th>
<th>The Reply Message Group Id, for example PR31_01sapserv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>device</td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
</tbody>
</table>

Description

Stores a SAP spool job notification for the specified device in the runtime repository. The notification states that the device is OK. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.
Note: If you need to include detailed information in the notifications you can use the `SAPDMDeviceNotificationSend`.

**Returns**
N/A

**Example**
```php
$notification = SAPDMDeviceOkSend($rmgid, $sapdevice);
```

### SAPDMJobErrorSend

**Syntax**
```php
SAPDMJobErrorSend($rmgid, $spoolid, $device);
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$rmgid</td>
<td>The Reply Message Group Id, for example PR31_01sapserv.</td>
</tr>
<tr>
<td>$spoolid</td>
<td>Internal spool Id. Required by SAP R/3 with RFC callback as the return parameter for identifying an SAP output request.</td>
</tr>
<tr>
<td>$device</td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
</tbody>
</table>

**Description**
Stores a SAP spool job notification for the specified spool job id in the runtime repository. The notification states that the job was erroneous. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

Note: If you need to include detailed information in the notifications you can use the `SAPDMJobNotificationSend`

**Returns**
N/A

**Example**
```php
$notification = SAPDMJobErrorSend($rmgid, $spoolid, $device);
```
SAPDMJobNotificationSend

Syntax

SAPDMJobNotificationSend(rmgid, spoolid, device, classcode, jobstatuscode, areacode, resultcode, unctime, language, reportlevel, message);

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rmgid</td>
<td>The Reply Message Group Id, for example PR31_01sapserv.</td>
</tr>
<tr>
<td>spoolid</td>
<td>Internal spool Id. Required by SAP R/3 with RFC callback as the return parameter for identifying an SAP output request.</td>
</tr>
<tr>
<td>device</td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
<tr>
<td>classcode</td>
<td>The class code. For available values, see the Return codes section of the Delivery Manager documentation.</td>
</tr>
<tr>
<td>jobstatuscode</td>
<td>The job status code. For available values, see the Return codes section of the Delivery Manager documentation.</td>
</tr>
<tr>
<td>areacode</td>
<td>The area code. For available values, see the Return codes section of the Delivery Manager documentation.</td>
</tr>
<tr>
<td>resultcode</td>
<td>The result code. For available values, see the Return codes section of the Delivery Manager documentation.</td>
</tr>
<tr>
<td>unctime</td>
<td>UNC date and time stamp in the following format: YYYYMMDDHHMMSS. For example, 20100228124500.</td>
</tr>
<tr>
<td>language</td>
<td>For example EN.</td>
</tr>
</tbody>
</table>
| reportlevel | To have an influence on performance you can control at which level notifications should be sent. 
01 - Completion
02 - Problem with intervention
03 - Problem without intervention
04 - Status change
05 - Information (no error)
09 - All available information
Note: A higher number includes the information sent in lower numbers. |
| message    | Detailed uninterpreted info. However logged by the SAP system. |
**Description**

Stores a SAP spool job notification for the specified spool job id in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

**Note:** This function allows you to specify detailed notification information. However, in most cases the `SAPDMJobOkSend` and `SAPDMJobErrorSend` functions can be sufficient to use.

**Returns**

N/A

**Example**

```php
(notification = SAPDMJobNotificationSend($rmgid, $sapspoolid, $sapdevice,"04", "04", "03", "01", ",","EN", "01", "Printed");
```

---

**SAPDMJobNotificationSendEx**

**Syntax**

`SAPDMJobNotificationSendEx($rmgid, $spoolid, $device, $occode);`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>rmgid</code></td>
<td>The Reply Message Group Id, for example PR31_01sapserv.</td>
</tr>
<tr>
<td><code>spoolid</code></td>
<td>Internal spool Id. Required by SAP R/3 with RFC callback as the return parameter for identifying an SAP output request.</td>
</tr>
<tr>
<td><code>device</code></td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
<tr>
<td><code>occode</code></td>
<td>An Output Center status code. The function translates this code to valid SAP status codes. See the Delivery Manager documentation for available Output Center status codes.</td>
</tr>
</tbody>
</table>

**Description**

Stores a SAP spool job notification for the specified spool job id in the runtime repository. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client.

This function allow you to specify an Output Center device status code as an input parameter, which the function translates to actual SAP parameters.

**Returns**

N/A
Example

```php
$notification = SAPDMJobNotificationSendEx($rmgid, $sasperspoolid, $sapdevice, "3");
```

SAPDMJobOkSend

**Syntax**

```php
SAPDMJobOkSend(rmgid, spoolid, device);
```

<table>
<thead>
<tr>
<th>rmgid</th>
<th>The Reply Message Group Id, for example PR31_01sapserv.</th>
</tr>
</thead>
<tbody>
<tr>
<td>spoolid</td>
<td>Internal spool Id. Required by SAP R/3 with RFC callback as the return parameter for identifying an SAP output request.</td>
</tr>
<tr>
<td>device</td>
<td>The device configured in SAP, i.e. the host name, for example DMPRINT.</td>
</tr>
</tbody>
</table>

**Description**

Stores a SAP spool job notification in the runtime repository. The notification states that the job was successfully printed. At a given interval, the Delivery Manager client reads the notifications in the repository and sends them to the SAP application server.

The interval used for scanning for notifications is the shortest interval of the intervals specified on the LOMS and on the DM Client

**Note:** If you need to include detailed information in the notifications you can use the `SAPDMDeviceNotificationSend`.

**Returns**

N/A

**Example**

```php
$notification = SAPDMJobOkSend($rmgid, $sasperspoolid, $sapdevice);
```
Sending file based notifications

For compatibility with existing StreamServe Projects, you can still use the job notification file and device notification file, i.e. if you do not want to do any manual changes in your Project.

Job notification file

Creating job and device notification files

Any application that can generate a tab-separated text file can be used to create the required job and device notification files. However, we recommend you use StreamServe to generate these files.

The format of the job notification file is defined by the XOM interface description and therefore no other types of data streams are allowed. The file contains fields with values that represent the current status.

Note: If you have a 4.0 Project before upgrading, you must create the new format for the job notifications, by either using the new function or patching the output.

Requirements

- A tab must separate each data field.
- Return values from the OMS that cannot be made available by the OMS must be represented by a dash.

Content

The content of the job notification file should contain the following fields, in this order:

<job_IdEx> <job_id><device><report_level> <class> <job_status> <area> <result> <UTC_time> <language> <message>

The <job_IdEx> field should be a combination of the job_id and the code sent from the OMS, to ensure that each job status is only reported once.

Note: There must be a tab between each field in the file.
Example 1  

A job status message in a job notification file

PR31_01sapserv.000003082500001.4 PR31_01sapserv.000003082500001 DMPRINT 01 04 04 03 01 
20020903130712 EN Message

<table>
<thead>
<tr>
<th>Field</th>
<th>Example OMS value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>job_IdEx</td>
<td>PR31_01sapserv.000003092600001.4</td>
<td>The file name is made up by the job Id and the OMS code.</td>
</tr>
<tr>
<td>job_Id</td>
<td>PR31_01sapserv.000003092600001</td>
<td>The job Id is made up by the RMG and the spool Id.</td>
</tr>
<tr>
<td>device</td>
<td>DMPRINT</td>
<td>The device configured in the SAP system. I.e. the host name.</td>
</tr>
<tr>
<td>report_level</td>
<td>01</td>
<td>Completion</td>
</tr>
<tr>
<td>class</td>
<td>04</td>
<td>Information</td>
</tr>
<tr>
<td></td>
<td>01</td>
<td>Error</td>
</tr>
<tr>
<td>job_status</td>
<td>04</td>
<td>Complete</td>
</tr>
<tr>
<td></td>
<td>08</td>
<td>Unknown</td>
</tr>
<tr>
<td>area</td>
<td>03</td>
<td>Formatting</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>Printing</td>
</tr>
<tr>
<td>result</td>
<td>01</td>
<td>Printed</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>Not printed</td>
</tr>
<tr>
<td>UTC_time</td>
<td>20020904171420</td>
<td>YYYYMMDDHHMMSS</td>
</tr>
<tr>
<td>language</td>
<td>EN</td>
<td>-</td>
</tr>
<tr>
<td>message</td>
<td>Test message</td>
<td>-</td>
</tr>
</tbody>
</table>

For other combinations of return codes, see Return codes on page 73.

For examples of how StreamServe can generate a job notification file, see Configuring the StreamServe Project on page 77.

File name

The name of the job notification file must contain the Reply Message Group (RMG), the SAP spool Id, and the OMS code. For example:
PR31_01sapserv.000003082500001.00123221

Note: No extension should be used in the file name.
**Destination**

The DM Client searches for the job notification file in the destination folder specified in the *Job notifications folder* option of the RFC Gateway DM Client.
Device notification file

Requirements
- A tab must separate each data field.
- Return values from the OMS that cannot be made available by the OMS must be represented by a dash.
- The device notification file must be a text file.

Content
The content of the device notification file should contain the following fields, in this specific order.
<deviceEx> <device> <report_level> <class> <UTC_time> <queue_enabled> <printing_enabled_alarm> <busy> <number_of_jobs> <incomplete_job_data>

Note: There must be a tab between each field in the file.

Example 2 A device status message in a device notification file

| STRSOMS.3 STRSOMS 04 04 20021010123000 Yes Yes Yes Yes . Yes |

The following is an example of the contents of a device notification file.

Device notification file fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Example OMS value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deviceEx</td>
<td>STRSOMS.3</td>
<td>The file name is made up by the device name and the OMS code.</td>
</tr>
<tr>
<td>device</td>
<td>STRSOMS</td>
<td>The device name</td>
</tr>
<tr>
<td>report_level</td>
<td>04</td>
<td>Status change</td>
</tr>
<tr>
<td>class</td>
<td>04</td>
<td>Information</td>
</tr>
<tr>
<td>UTC_time</td>
<td>20021106134230</td>
<td>YYYYMMDDHHMMSS</td>
</tr>
<tr>
<td>queue_enabled</td>
<td>X</td>
<td>Yes</td>
</tr>
<tr>
<td>printing_enabled_alarm</td>
<td>-</td>
<td>No</td>
</tr>
<tr>
<td>busy</td>
<td>X</td>
<td>Yes</td>
</tr>
<tr>
<td>number_of_jobs</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>incomplete_job_data</td>
<td>-</td>
<td>No</td>
</tr>
</tbody>
</table>

For other combinations of return codes, see Return codes on page 73.
For examples of how StreamServe can generate a device notification file, see *Configuring the StreamServe Project* on page 77.

**File name**

The name of the device notification file must be the same as the host name parameter defined for the device in the SAP system, combined with the OMS code. For example, STRSOMS.12323.

**Note:** No extension should be used in the file name.

**Destination**

The DM Client searches for the device notification file in the device folder specified in the *Device notifications folder* option on the RFC Gateway DM Client.
Return codes

This chapter lists return codes that are used when creating job and device notification files, see Sending notifications to SAP on page 55.
**Job status codes**

Although the OMS system can have its own sets of error and status codes, the return codes sent to the SAP system must always map to the ones listed below. The job notification file is mapped to different code combinations depending on the status of the job. You must ensure that the return codes sent back are not contradictory.

The following table shows the job status codes that can be used as content for the job notification file, see *Job notification file* on page 68.

**Note:** The codes are translated by SAP and can vary from the descriptions in the table.

<table>
<thead>
<tr>
<th>Code</th>
<th>OMS value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td>01</td>
<td>Error</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>Problem requiring intervention</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>Problem not requiring intervention</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>Information (no error)</td>
</tr>
<tr>
<td>job_status</td>
<td>01</td>
<td>Pre-processing (formatting)</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>Pending (waiting in queue)</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>Processing (printing)</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>Complete (cannot be resubmitted)</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>Retained (complete but still stored within the OMS)</td>
</tr>
<tr>
<td></td>
<td>06</td>
<td>Cancelled</td>
</tr>
<tr>
<td></td>
<td>07</td>
<td>Gone (no job information)</td>
</tr>
<tr>
<td></td>
<td>08</td>
<td>Unknown (probably bad job Id)</td>
</tr>
<tr>
<td>area</td>
<td>01</td>
<td>Spooler</td>
</tr>
<tr>
<td></td>
<td>02</td>
<td>Printing</td>
</tr>
<tr>
<td></td>
<td>03</td>
<td>Formatting</td>
</tr>
<tr>
<td></td>
<td>04</td>
<td>Connection (network)</td>
</tr>
<tr>
<td></td>
<td>05</td>
<td>Other</td>
</tr>
</tbody>
</table>
Job status codes

<table>
<thead>
<tr>
<th>Code</th>
<th>OMS value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>result</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>Printed</td>
</tr>
<tr>
<td>02</td>
<td>02</td>
<td>Not printed</td>
</tr>
<tr>
<td>03</td>
<td>03</td>
<td>Partly printed</td>
</tr>
<tr>
<td>04</td>
<td>04</td>
<td>Possibly printed</td>
</tr>
<tr>
<td>05</td>
<td>05</td>
<td>Output changed</td>
</tr>
</tbody>
</table>

Example 3  Code combination for a job completed successfully

PR31_01sapserv.000003082500001 DMPRINT 04 04 03 01 20020903130712 EN Message\text

Example 4  Code combination for a failed job

PR31_01sapserv.000003082500001 DMPRINT 01 06 03 02 20020903130712 EN Message\text
Device status codes

Although an OMS system can have its own sets of error and status codes, the return codes sent to the SAP system must always map to the ones listed below. The device status file is mapped to different code combinations depending on the status of the device. You must ensure that the return codes sent back make sense.

The following table shows the device status codes that can be used as content for the device notification file, see Device notification file on page 71.

<table>
<thead>
<tr>
<th>Code</th>
<th>OMS value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>class</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td>Error</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>Problem requiring intervention.</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td>Problem not requiring intervention.</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td>Information (no error)</td>
<td></td>
</tr>
<tr>
<td>UTC_time</td>
<td>YYYYMMDDHHMMSS</td>
<td>Date and time</td>
</tr>
<tr>
<td>queue_enabled</td>
<td>Yes / No</td>
<td>-</td>
</tr>
<tr>
<td>print_enabled</td>
<td>Yes / No</td>
<td>-</td>
</tr>
<tr>
<td>printing_enabled_alarm</td>
<td>Yes / No</td>
<td>-</td>
</tr>
<tr>
<td>busy</td>
<td>Yes / No</td>
<td>-</td>
</tr>
<tr>
<td>number_of_jobs</td>
<td>N</td>
<td>Where N indicates the number of jobs.</td>
</tr>
<tr>
<td>incomplete_job_data</td>
<td>Yes / No</td>
<td>-</td>
</tr>
</tbody>
</table>

Example 5  
Code combination for an error free device

04 20021010123000  Yes Yes Yes Yes . Yes

Example 6  
Code combination for a device in error state

01 20021010123000  Yes Yes Yes Yes . Yes
Configuring the StreamServe Project

There are several ways you can configure a StreamServe Project to produce notifications for job status and device status. The recommended way is to use the built-in StreamServe script functions. There are a number of Delivery Manager script functions allowing you to store notifications in the runtime repository.

Note: The SP5 version of the connectivity pack is backwards compatible with Design Center Projects without need for any manual changes.

Scripting

The StreamServe installation includes a number of built-in StreamServe script functions. These functions can be called from a StreamServe Project to produce job notifications and device notifications for intermediate storage in the runtime repository.

Scripts are run at specific times and phases during the execution of a StreamServe Project. You can execute scripts before and after Messages and Processes. After an Event has been retrieved (including its fields and variables), you can execute Event Retrieved scripts.

The time you choose to run a script depends on the action you want and the preferred result. In this example the script has been placed as an After script on the Job. See Configuring the Runtime on page 78.

For more details, see the Streamserve Scripting Reference.

Input connector

If you run the DM Server/ DM Client level of integration, you can use the Service Request input connector for receiving input from SAP.

For a DM Command / DM Client integration, you normally use a Directory connector to receive files from SAP.

Configuring a StreamServe Project

You can create a new Project, or apply this configuration to an existing Project. For information about Projects, see the standard StreamServe documentation.
Configuring the Runtime

For the Delivery Manager, the Runtime is configured to run a script that calls the built-in Delivery Manager script functions. In the procedure below, the script is placed as an After script on the Job. You can place the script as appropriate for your Project.

If you use the DM Server and a Service Request input connector where you have specified a XOM Service name, you can use the default variant of the functions that store job notifications in the runtime repository. For example, the `SAPDMDefaultJobOkSend()` function, which uses the SAP metadata directly as input parameters.

There are no additional settings required for the Delivery Manager Project in the Runtime configuration.

**To configure the runtime Job for notifications with default values from SAP metadata using Service Request connector**

1. In the Runtime configuration, add a script to the Job.
2. Add one or more script functions to the After script to store notifications in the runtime repository. For details, see the *StreamServe Scripting Reference*.

![Figure 1 Script storing job and device status with default values.](image)

**To configure the runtime Job for notifications including Output Center status codes**

1. In the Runtime configuration, add a script to the Job.
2. Add the following variables to the After script:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$spoolid</code></td>
<td>The SAP spool Id is a unique Id that follows the spool job during the complete process. You retrieve the Id with: <code>$spoolid = GetConnectorValue(&quot;S_SPOOLID&quot;);</code></td>
</tr>
</tbody>
</table>
3 Add one or more script functions to the After script to store notifications in the runtime repository. For details, see the StreamServe Scripting Reference.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$device</td>
<td>The incoming file name has the format rmg.sapspoolid_device. You retrieve the device with: $device = GetConnectorValue(&quot;OSPRT&quot;);</td>
</tr>
<tr>
<td>$rmgid</td>
<td>You retrieve the Reply Message Group Id with: $rmgid = GetConnectorValue(&quot;RMG_ID&quot;);</td>
</tr>
</tbody>
</table>

Figure 2  Script storing job and device status, including Output Center status code, in runtime repository.

The job and device status mappings used with Output Center

The following tables describe the job and device status mappings done by submitting the $occode parameter to the SAPDMJobNotificationSendEx and SAPDMDeviceNotificationSendEx script functions, and are used with the Output Center. You can use this example as a template if you want to implement an OMS other than the Output Center.

<table>
<thead>
<tr>
<th>Job status code mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC code</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>
### Job status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Job status</th>
<th>Area</th>
<th>Result</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>04</td>
<td>Queued</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>04</td>
<td>Printing</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>01</td>
<td>Printed</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>01</td>
<td>Error</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>02</td>
<td>Partially printed</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>03</td>
<td>Preprint command error</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>03</td>
<td>Data stream not supported</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>04</td>
<td>Cancel</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>01</td>
<td>Submission error</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>04</td>
<td>Sent to printer</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>02</td>
<td>Server terminated while printing</td>
</tr>
</tbody>
</table>

### Device status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Queue</th>
<th>Printer</th>
<th>Alarm</th>
<th>Busy</th>
<th>Incomplete</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>01</td>
<td>Unknown</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Ready</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>04</td>
<td>Printing</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Out of paper</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Out of toner</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Jammed</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>03</td>
<td>Maintenance</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>01</td>
<td>Other</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>01</td>
<td>Unreachable</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>01</td>
<td>Protocol not supported</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Online</td>
</tr>
</tbody>
</table>
### Device status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Queue</th>
<th>Printer</th>
<th>Alarm</th>
<th>Busy</th>
<th>Incomplete</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Offline</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Warming up</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>03</td>
<td>Toner low</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Cancelling job</td>
</tr>
<tr>
<td>45</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>04</td>
<td>Processing job</td>
</tr>
<tr>
<td>50</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>04</td>
<td>Receiving job</td>
</tr>
<tr>
<td>51</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Received job</td>
</tr>
<tr>
<td>55</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Operator intervention</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Cover open</td>
</tr>
<tr>
<td>65</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Paper jam</td>
</tr>
<tr>
<td>70</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Toner low - press go</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Offline</td>
</tr>
<tr>
<td>80</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Load paper</td>
</tr>
<tr>
<td>85</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>MP tray load letter</td>
</tr>
</tbody>
</table>
Configuring the Platform

Input connector

You use a Service Request input connector where you specify a Service name. When you start the StreamServer application deployed with your Project configuration, this application starts and waits for output files from SAP. When you start the RFC Gateway where you have configured a DM Server, the DM Server can send files to the Service Request service.

To configure a Service Request input connector

1. Right-click the platform and select **New Input Connector**.
2. Double-click the connector to open the settings dialog.
3. From the **Selected layer** drop-down list, select **Platform (Physical)**.
4. From the **Connector type** drop-down list, select **Service Request**.
5. In the Request type field, select **XOM**.
6. In the Service name field, enter a Service name.
   
   **Note**: This name must be identical to the Service name you have specified on the RFC Gateway DM Server.

7. From the Selected layer drop-down list, select the platform.
8. From the Queue drop-down list, select **Input**.
9. Click **OK**.

![Figure 3 Service Request connector settings](image)

Output connector

As the Delivery Manager uses script functions to create notifications, a separate output connector is not required in the Platform. You can either use an existing output connector defined for your Message, or you can create a new Null output connector and specify the connector as a dummy connector.
Enable automatic sending of notifications back to SAP

For Projects where you use the Service Request input connector and the DM Server, you can enable automatic sending of status notifications via the DM Client to SAP, without scripting in the Design Center Project. The top-job status is returned allowing errors occurring during asynchronous delivery of documents to be caught.

To enable automatic sending of status notifications

1. In Design Center, select **Edit > Project Export Settings**. The Project Export Settings dialog opens.
2. Click the **Notifications** tab and select **Enable XOM Notifications**.

**Note:** This requires that the ExtJobID of the top-job is set to the following:

```
SAP:<rmg_id>:<spool_id>:<device>
```

The DM Server sets this automatically when a job is submitted via the Service Request connector. Alternatively, you can run the `setExtJobId` script function at the beginning of the job with the following argument:

```
"SAP:"+$rmgid+."+$spoolid+."+$device
```
Configuring the Platform

Configuring the StreamServe Project
Configuring the Project for file based notifications

You can configure a Design Center Project to produce content for job and device notification files. For example, when halting Messages you must use file based notifications. Prior to Persuasion SP5, file based notifications was the only possibility.

The notification files are used by the Delivery Manager to update the SAP system with status information. This chapter describes one example of how to configure a StreamServe Project.

Note: This guide only contains instructions specific to configuring StreamServe for the Delivery Manager solution. For general information about configuring StreamServe, see the standard StreamServe documentation.

For the Deliver Manager, there are no specific settings required for the Platform.

Included activities

- Using the strsdm.fcn function file on page 86
- Configuring a StreamServe Project on page 86
- Adding the strsdm.fcn function file to a Resource set on page 86
- Configuring the Runtime on page 87
Using the strsdm.fcn function file

The Delivery Manager installation includes a function file, strsdm.fcn. This file contains script functions that can be called from a StreamServe Project to produce the job notification file and the device notification file.

Scripts are run at specific times and phases during the execution of a StreamServe Project. You can execute scripts before and after Messages and Processes. After an Event has been retrieved (including its fields and variables), you can execute Event Retrieved scripts.

The time you choose to run a script depends on the action you want and the preferred result. In this example, the script has been placed as an After script on the Job. See Configuring the Runtime on page 87.

The strsdm.fcn function file also includes mappings from StreamServe Output Center status codes to SAP job and device notification codes. For more information, read the notes included in the function file.

Configuring a StreamServe Project

You can create a new Project, or apply this configuration to an existing Project. For information on Projects, see the standard StreamServe documentation.

Adding the strsdm.fcn function file to a Resource set

In order for StreamServer to access the strsdm.fcn function file, you must add the function file to the Resource Set in your StreamServe Project. You import the strsdm.fcn function file into the Resource Set in the Design Center.

To import the strsdm.fcn function file into a Resource Set

1. In your StreamServe Project, create or open the Resource Set, to which you want to import the strsdm.fcn function file.
2. In the Resource Set, right-click the function files folder and select Import.
3. Browse to, and select the strsdm.fcn function file from the Delivery Manager etc directory. For example:

   \StreamServe\Applications\SAPConnect\5.5.0\sapdm\etc

4. Click Open. The Select Custom folder opens.
5. Select the appropriate folder.
6. Click OK. The function file is added as a new resource to the folder.
Configuring the Runtime

For the Delivery Manager, the Runtime is configured to run a script that calls the `strsdm.fcn` function call. In this example, the script is placed as an After script on the Job. You can place the script as appropriate for your Project.

There are no additional settings required for the Delivery Manager Project in the Runtime configuration.

**To add a function call to a Job in the Runtime**

1. In the Runtime configuration, add a script to the Job.
2. Add the following variables to the After script:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$sapspoolid</td>
<td>The SAP spool Id is a unique Id that follows the spool job during the complete process. The incoming file name has the format <code>rmg.sapspoolid_device</code>. The SAPDMGetSpoolIdFromFileName function is used to extract the $sapspoolid from the file name.</td>
</tr>
<tr>
<td>$devname</td>
<td>The incoming file name has the format <code>rmg.sapspoolid_device</code>. The SAPDMGetDeviceIdFromFileName function is used to extract the $devname from the file name. For more information, see the StreamServe Connect for SAP - Output+ and StreamServe Connect for SAP - E-docs documentation.</td>
</tr>
<tr>
<td>$time</td>
<td>The SAPDMUNCTime function creates the time format required by the SAP system. For example. $time= SAPDMUNCTime ();</td>
</tr>
</tbody>
</table>

3. Add the `SAPDMJobNotificationOutEx` function call.

**Included topics**

- `SAPDMJobNotificationOutEx function call` on page 88
- `Configuring the Platform` on page 91
SAPDMJobNotificationOutEx function call

The SAPDMJobNotificationOutEx function sets values to the job notification file. The function accepts the following arguments:

- #path
- #spoolid
- #device
- #ocjcode

Example 7 SAPDMJobNotificationOut function call

SAPDMJobNotificationOutEx("D:\strs\notify", $sapspoolid, $sapdevice, "3");

This results in a Completed status in the SAP system.

SAPDMDeviceNotificationOutEx function call

The SAPDMDeviceNotificationOutEx function sets values to the device notification file. The function accepts the following arguments:

- #path
- #devname
- #ocdcode

Example 8 SAPDMJobNotificationOut function call

SAPDMDeviceNotificationOutEx("D:\strs\device", $devname, "1");

This results in a Ready status of the printer.

The job and device status mappings used with the Output Center

The following tables describe the job and device status mappings included in the strsdm.fcn file. These mappings are done by the SAPDMJobNotificationOut and SAPDMDeviceNotificationOut function calls, and are used with Output Center. You can use this example as a template if you want to implement an OMS other than the Output Center.
### Job status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Job status</th>
<th>Area</th>
<th>Result</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>01</td>
<td>Unknown</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>04</td>
<td>Queued</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>04</td>
<td>Printing</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>01</td>
<td>Printed</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>01</td>
<td>Error</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>02</td>
<td>Partially printed</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>03</td>
<td>Preprint command error</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>03</td>
<td>Data stream not supported</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>04</td>
<td>Cancel</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>2</td>
<td>01</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>01</td>
<td>Submission error</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>04</td>
<td>Sent to printer</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>02</td>
<td>Server terminated while printing</td>
</tr>
</tbody>
</table>

### Device status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Queue</th>
<th>Printer</th>
<th>Alarm</th>
<th>Busy</th>
<th>Incomplete</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>01</td>
<td></td>
<td>Unknown</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>04</td>
<td></td>
<td>Ready</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td>04</td>
<td></td>
<td>Printing</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>Out of paper</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>Out of toner</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>02</td>
<td></td>
<td>Jammed</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>03</td>
<td></td>
<td>Maintenance</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>01</td>
<td></td>
<td>Other</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>01</td>
<td></td>
<td>Unreachable</td>
</tr>
</tbody>
</table>
## Device status code mappings

<table>
<thead>
<tr>
<th>OC code</th>
<th>Class</th>
<th>Queue</th>
<th>Printer</th>
<th>Alarm</th>
<th>Busy</th>
<th>Incomplete</th>
<th>Report level</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>3</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>01</td>
<td>Protocol not supported</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Online</td>
</tr>
<tr>
<td>25</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Offline</td>
</tr>
<tr>
<td>30</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>04</td>
<td>Warming up</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>03</td>
<td>Toner low</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Cancelling job</td>
</tr>
<tr>
<td>45</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>04</td>
<td>Processing job</td>
</tr>
<tr>
<td>50</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Receiving job</td>
</tr>
<tr>
<td>51</td>
<td>4</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Received job</td>
</tr>
<tr>
<td>55</td>
<td>3</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>02</td>
<td>Operator intervention</td>
</tr>
<tr>
<td>60</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Cover open</td>
</tr>
<tr>
<td>65</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Paper jam</td>
</tr>
<tr>
<td>70</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Toner low - press go</td>
</tr>
<tr>
<td>75</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>04</td>
<td>Offline</td>
</tr>
<tr>
<td>80</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>Load paper</td>
</tr>
<tr>
<td>85</td>
<td>2</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>02</td>
<td>MP tray load letter</td>
</tr>
</tbody>
</table>
Configuring the Platform

Input connector
For Delivery Manager, you need to configure the input connector to scan the directory to where Delivery Manager copies the SAP output.

Output connector
As the Delivery Manager uses a function to create an output file, an output connector is not required in the Platform. You can either use an existing output connector defined for your Message, or you can create a new Null output connector and specify the connector as a dummy connector.
Configuring the Platform

Configuring the Project
Testing the Delivery Manager configuration

Before you use the Delivery Manager, you should test the configuration.

Prerequisites

Before testing the Delivery Manager configuration, you must have installed and configured the SAP Configuration, the Delivery Manager configuration and the StreamServe Project.

Included activities

- Testing the DM Server and Client on page 94
- Testing the DM Command on page 96
Testing the DM Server and Client

To test the DM Server and DM Client, you can:

- Start the RFC Gateway.
- Test the connection.
- Output a device list and verify.

**To start the RFC Gateway**

1. Start Control Center.
2. In Control Center, start the RFC Gateway, see *Starting the DM Server* on page 38.
3. Verify that it has started.

**To test the DM Server connection to the SAP system**

1. Logon to your SAP system.
2. In the transaction box, enter /nsm59 to locate the RFC destination you have created.
3. Click **Test Connection** to verify the connection. If no error messages occur, the configuration is correctly defined.

**To output a device list and verify**

1. Output a device list:
   - In the transaction box, enter /nspad.
   - Select **Output devices** > **Print this list**.
   - Select the device configured for the Delivery Manager, and verify the device settings.
   - Click **Continue**.
2. Verify that the SAP spool file was copied by the DM Server to the target destination. The target destination is specified in the RFC Gateway / DM Server.
3. If you use the built-in script functions to create job notifications, verify in the runtime repository that the notifications have been stored.
4 If you use notification files, verify that StreamServer produced the job notification file, and that the file was sent to the Notification folder. The Notification folder is specified in **Job notifications folder** option in the DM Client settings.

5 In the SAP system, enter `/nsp02` in the transaction box to verify the job status in the SAP system.

   If the Delivery Manager successfully reported the job notification, the job has a **Complete** status.
Testing the DM Command

To test the DM Command

1. Logon to your SAP system.
2. Output the device list:
   a. In the transaction box, enter /nspad.
   b. Select Output devices > Print this list.
   c. Select the device configured for the Delivery Manager, and verify the device settings.
3. In Control Center, view the log file displayed in the Log window to verify that DM Submit received the job.
4. Verify that StreamServer produced the job notification file, and that the file was sent to the notification folder or the notifications table in the runtime repository. If you use a job notification folder, it is specified in the Job notification folder option on the DM Client.
5. Verify that StreamServer produced the device notification file, and that the file was sent to the notification folder or the notifications table in the runtime repository. If you use a device notification folder, it is specified in the Device notification folder option on the DM Client.
6. In the SAP system, enter /nsp02 in the transaction box to verify the job status in the SAP system.

   If the Delivery Manager successfully reported the job notification, the job has a Complete status.

<table>
<thead>
<tr>
<th>DM Command - Submit settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Print Immediately</td>
</tr>
<tr>
<td>New Spool request</td>
</tr>
<tr>
<td>Delete after output</td>
</tr>
</tbody>
</table>

   d. Click Continue.
Troubleshooting

This section provides information about how to solve problems which can occur when using the DM Server and Client, or DM Command.

**DM Server and Client**
- *Errors when submitting a job* on page 98

**DM Command**
- *ROMS not available during SAP configuration* on page 99
- *Errors when submitting a job* on page 99

**UNIX**
- *Errors when running Delivery Manager applications in UNIX* on page 100.
Troubleshooting DM Server and Client

Included scenarios

- Errors when submitting a job on page 98

Errors when submitting a job

When you have submitted a job using the DM Server, and an error message stating “Failed to connect to host spooler” appears in the Output Controller (transaction code /nsp02), you can do the following:

1. Verify that the DM Server has started and re-send the job, see Starting the DM Server on page 38.

2. Verify that the correct RFC destination has been specified as the Tasking Target value for the LOMS. See Configuring a Logical Output Management System (LOMS) on page 15.

3. Verify that the Program ID name specified on the DM Server task in the RFC Gateway is the same as the Program Id defined for the RFC destination. (The Program Id name is case sensitive.)
   - In your SAP system, enter /nsm59 in the transaction box.
   - Search for the RFC destination you created.
   - Verify that the Program ID name is specified on the DM Server task in the RFC Gateway.

DM Client cannot process notifications and stops

1. Verify that the notification file format is valid.

2. If the problem persists with valid notification files, manually remove the notifications from the notification table in the runtime repository or from the notification folders.
Troubleshooting DM Command

Included scenarios

- ROMS not available during SAP configuration on page 99
- Errors when submitting a job on page 99

ROMS not available during SAP configuration

When you are creating a LOMS, and you want to select the ROMS but it is not available in the list, you can refresh the SAP configuration.

To refresh a SAP configuration to access a ROMS

1. In the transaction box, enter /nspad.
2. Go back to the Create LOMS view. This refreshes the configuration and the ROMS is available.

Errors when submitting a job

When you have submitted a job using the Command Line interface, and an error message displays stating "Source Path definition not found", you can check the following:

1. Verify that the strsdmsubmit.log file was created. If not, check the paths to the Delivery Manager working directory (\strsoms), and the paths to the strsdmsubmit application.
2. Open the strsdmsubmit.log file. If the source error is shown in the SAP system, the last line in the strsdmsubmit.log file will show:
   strsdmsubmit:2.00 1 5 - Source\ path\ definition\ not\ found
3. In the transaction box, enter /nspad.
4. Select Utilities > Display SAPPARAM. The Important Profile Parameters window opens. The path on the rspo/to_host/datafile line is the value to use for the source.31 parameter in the strsdmstart.cfg file.
5. Check that the correct system name is specified in the strsdmstart.cfg file.
6. Submit the job again and verify that the same error message is not displayed.
Errors when running Delivery Manager applications in UNIX

In UNIX, if the Delivery Manager applications do not start and log files are not generated, you can check:

- That the `strsdmclient` and `strsdmsubmit` executables have full user executable rights.
- That the SAP user has full access to the folders and files used when running the Delivery Manager. We recommended that you prohibit access on world level.
Useful SAP transaction codes

This section lists SAP transaction codes which are commonly used in the SAP system to activate transactions.

Note: To enter a transaction code from any screen within the SAP system other than the initial screen, prefix the code with /n. For example, the /nVF03 transaction code displays the Display Billing Document screen from any screen in the SAP system.

Configuration

- SM59: Display and maintain RFC destinations
- SPRO: Customizing
- OMFE: Processing Program/Layout Set for Purchase Order (MM)
- V/30: Processing Program/Layout Set for Order Confirmation (SD)
- V/34: Processing Program/Layout Set for Delivery Note (SD)
- V/40: Processing Program/Layout Set for Invoice (SD)

Spool functions

- SPAD: Spool Administration
- SP01: Spool Requests

Form processing

- SE71: SAPscript
- SE73: Font Maintenance
- SE78: Graphics Management
- SMARTFORMS: Smart Forms
- SO10: Standard Texts
Useful SAP transaction codes

Programs and reports

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE38</td>
<td>ABAP Editor</td>
</tr>
<tr>
<td>RSTXSCRIP</td>
<td>Import/Export SAPscript objects and XOM configuration</td>
</tr>
<tr>
<td>RSTXSYMB</td>
<td>List SAP symbols</td>
</tr>
<tr>
<td>RSTXICON</td>
<td>List SAP icons</td>
</tr>
<tr>
<td>RSPO0049</td>
<td>Activate Access Method Z (Spool Exit)</td>
</tr>
</tbody>
</table>

Generating application output

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB90</td>
<td>Print Purchase Order (MM)</td>
</tr>
<tr>
<td>VA02</td>
<td>Change Sales Order (Order Confirmation, SD)</td>
</tr>
<tr>
<td>VA03</td>
<td>Display Sales Order (Order Confirmation, SD)</td>
</tr>
<tr>
<td>VF02</td>
<td>Change Billing Document (Invoice, SD)</td>
</tr>
<tr>
<td>VF03</td>
<td>Display Billing Document (Invoice, SD)</td>
</tr>
<tr>
<td>VL02</td>
<td>Change Outbound Delivery (Delivery Note, SD)</td>
</tr>
<tr>
<td>VL03</td>
<td>Display Outbound Delivery (Delivery Note, SD)</td>
</tr>
<tr>
<td>SM69</td>
<td>List of external commands (for box drawing characters)</td>
</tr>
<tr>
<td>SM04</td>
<td>List of users currently logged on (short list)</td>
</tr>
</tbody>
</table>

Data and metadata

<table>
<thead>
<tr>
<th>Transaction Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SE11</td>
<td>Data Dictionary</td>
</tr>
<tr>
<td>SE16</td>
<td>Data Browser</td>
</tr>
<tr>
<td>WE63</td>
<td>IDoc Types</td>
</tr>
</tbody>
</table>
## Glossary

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DM</td>
<td>Delivery Manager</td>
</tr>
<tr>
<td>LOMS</td>
<td>Logical Output Management System</td>
</tr>
<tr>
<td>OMS</td>
<td>Output Management System</td>
</tr>
<tr>
<td>RFC</td>
<td>Remote Function Call</td>
</tr>
<tr>
<td>ROMS</td>
<td>Real Output Management System</td>
</tr>
<tr>
<td>XOM</td>
<td>X/Open OSI-Abstract-Data Manipulation interface</td>
</tr>
</tbody>
</table>