Application Server

Release 2015
Disclaimer
This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes only.

Sample Code Warranty disclaimer
Microsoft Corporation disclaims any warranty regarding the sample code contained in this documentation, including the warranties of merchantability and fitness for a particular purpose.

License agreement
Use of this software is covered by a license agreement provided with it. If you have any questions, please call the Customer Assistance Department at 800-456-0025 (in the United States or Canada) or +1-701-281-6500.

Copyright
© 2014 Microsoft Corporation. All rights reserved.

Publication Date
September 2014
Contents

Introduction 1

Application Server Module Overview .......................................................... 1
Application Server Hardware Environment ............................................... 2
Operating Sequence ...................................................................................... 2
User Guide Overview .................................................................................. 3
What is Covered in the User Guide? ............................................................ 3
Who Should Use the User Guide? ................................................................. 3
How to Use This Documentation .................................................................. 3
Quick Reference Task List .......................................................................... 4
How Do I Set Up...? ...................................................................................... 4
How Do I Launch...? ..................................................................................... 4
How Do I Check...? ...................................................................................... 4
How Do I Change...? ................................................................................... 4
How Do I Delete...? .................................................................................... 4
How Do I Pause...? ..................................................................................... 4
How Do I Restart...? .................................................................................... 4
How Do I Shut Down...? .............................................................................. 4
How Do I Submit...? .................................................................................... 4
How Do I Resubmit...? ................................................................................. 4

Administering the Application Server 5

Overview ..................................................................................................... 5
Setting Up an Application Server ............................................................... 6
Starting Application Server ....................................................................... 14
Checking the Application Server Queue ..................................................... 15
Viewing Application Server Logs ............................................................... 17
  Viewing Activity for a Specific Request ..................................................... 17
  Viewing the Entire Activity Log ................................................................ 18
Pausing, Restarting, and Shutting Down Application Servers ..................... 19
  Pausing Application Servers ..................................................................... 19
  Restarting Application Servers ................................................................. 19
  Shutting Down Application Servers .......................................................... 19
Using Alternate Methods ........................................................................... 19
  Viewing Information about All Application Servers .................................. 20
Changing Application Server Properties and Options ............................... 22
Deleting an Application Server ................................................................... 23

Using Application Server 25

Overview ..................................................................................................... 25
  Types of Requests ...................................................................................... 25
Submitting Requests .................................................................................. 27
Submitting Requests Via Email ................................................................. 32
Resubmitting Incomplete Requests ........................................................... 34
Deleting Requests from the Application Server Queue ............................... 35
Processing Requests Created by Quick Send .............................................. 36
Processing Requests Created by Doc Share .............................................. 37

Reference 39

Application Server Module Menu ............................................................... 39
Application Server (96.010.00)................................................................... 40
  Application Server, Processing Log Tab .................................................... 41
  Application Server, Queue Status Tab ........................................................ 42
Submit to Application Server (96.020.00)................................................................. 43
  Submit to Application Server, General Tab ............................................................... 45
  Submit to Application Server, Scheduling Tab ......................................................... 46
  No Scheduling ............................................................................................................. 47
  One Time Scheduling Frequency ............................................................................... 48
  Daily Scheduling Frequency ...................................................................................... 49
  Weekly Scheduling Frequency ................................................................................... 50
  Monthly Scheduling Frequency .................................................................................. 51
  Event-Based Scheduling .............................................................................................. 52
  Submit to Application Server, Report Routing Tab ..................................................... 53
  Submit to Application Server, Advanced Tab ............................................................ 54

Application Server Status (96.030.00)......................................................................... 55
  Grid/Form (when Current Queue option is selected) .................................................. 56
  Grid/Form (when Master Schedule option is selected) .............................................. 58
  Grid/Form (when Today's Schedule option is selected) .............................................. 60

Application Server Status (96.03.010)........................................................................ 62
  Application Server Status, Companies Tab ................................................................. 63
  Application Server Status, Request Log Tab ............................................................... 64
  Application Server Status, Schedule Information Tab ................................................. 65

Application Server Request Log (96.060.00)............................................................... 66

Application Server Administration Wizard (96.070.00) ........................................... 67

Application Server Administration (96.080.00).......................................................... 68

Application Server Properties ...................................................................................... 69
  Application Server Properties, General Tab ............................................................... 70
  Application Server Properties, Management Tab ...................................................... 72
  Application Server Properties, User Assignments Tab ............................................. 73
  Application Server Properties, Default Server Tab .................................................... 74

Developer’s Guide 77

Introduction .................................................................................................................. 77

About This Developer’s Guide ....................................................................................... 78
  Setting Up the Developer Objects .............................................................................. 78
  System Database Connectivity and Object Initialization .......................................... 78
  Accessing Application Server Functions .................................................................... 78
  Developer’s Reference ............................................................................................... 78

Setting Up the Developer Objects ................................................................................. 79
  Setting Up the Objects for Visual Basic ...................................................................... 79
  Setting Up the Objects for VBA ................................................................................. 80

System Database Connectivity and Object Initialization .............................................. 81
  Establishing Database Connection Information ....................................................... 81
  Initializing the Developer Objects .............................................................................. 82

Accessing Application Server Functions ....................................................................... 83
  Submitting an Application Server Request .............................................................. 85
  Submitting a Request with Scheduling Information .................................................. 86
  Viewing Requests in the Queue ................................................................................ 90
  Displaying Processing Log Information .................................................................... 91
  Viewing Administrative Information .......................................................................... 92
  Upload a document using DocShare settings .......................................................... 93
  Upload a document to SharePoint Document Library ............................................ 94

Developer’s Reference ................................................................................................ 95
  Asdevobj.dll and Microsoft.Dynamics.SLAS.DevObj.dll ............................................ 96
  Schedule.dll and Microsoft.Dynamics.SLAS.Schedule.dll ..................................... 108
  Microsoft.Dynamics.SL.Communications ................................................................. 112

Glossary 113

Index 115
Introduction

Application Server Module Overview

The Application Server module in Microsoft Dynamics® SL relieves client workstations of extreme processing loads by enabling users to offload time-consuming processes and reports from their client workstations to separate server PCs running Application Server. After submitting a process or report request to the Application Server request queue, users can return immediately to normal operations. The Application Server checks the queue for outstanding requests and runs each process or report according to its priority in the queue as if the process or report had been run at a client workstation.

You can use a computer that is running Application Server and Quick Send to transmit the following documents electronically to customers, vendors, and employees:

- Invoices and statements created in the Accounts Receivable module
- Invoices and construction billings produced in the Flexible Billings module
- Invoices created in the Service Dispatch, and Service Contracts module
- Order confirmations, manual order confirmations, shipping notices, and invoices produced in the Order Management module
- Purchase orders created in the Purchasing module
- Direct deposit advice slips generated in the Payroll module

In addition, you can use Application Server and Doc Share to publish customer, vendor, or project documents on a SharePoint site for ease of access. The types of documents that you can post on a SharePoint site are:

- Invoices and statements created in the Accounts Receivable module
- Invoices and construction billings produced in the Flexible Billings module
- Order confirmations, manual order confirmations, and invoices produced in the Order Management module
- Purchase orders created in the Purchasing module

Providing true three-tier client/server functionality, the Application Server module works in all Microsoft Dynamics SL environments. Because it is easily scalable, Application Server can operate successfully in Microsoft Dynamics SL environments of all sizes. You can set up and simultaneously operate as many Application Servers as necessary for your environment. Application Servers can be started and stopped in any combination to match workload requirements at any given time.
Application Server Hardware Environment

Set up the Application Server hardware environment to suit your business needs. Typically, each Application Server operates on a server PC that is separate from, but linked to, the server PC operating the database.

![Diagram of typical Application Server environment]

Figure 1: Typical Application Server environment

Operating Sequence

When using the Application Server module, the general sequence of tasks is:

1. Set up an Application Server.
2. Launch the Application Server. The Application Server polls the queue for requests.
3. Submit Microsoft Dynamics SL process and report requests to the Application Server queue. The previously launched Application Server is already polling the Application Server queue. As the requests are submitted, the Application Server automatically begins running the requests.
User Guide Overview

This user guide provides administrators with task-oriented and reference information for the Application Server. Reviewing the user guide helps in making informed decisions regarding the implementation of the Application Server module in your business.

What is Covered in the User Guide?

The user guide consists primarily of procedures and checklists that show how to perform various tasks featured in the Application Server module. The user guide also contains topics that help you become better acquainted with the capabilities of the module. Topics are arranged in a logical order that builds on information previously presented in other Microsoft Dynamics SL user guides.

Who Should Use the User Guide?

The user guide is designed for System Administrators who are new to Microsoft Dynamics SL. The guide provides the information necessary for making decisions regarding how to use the Application Server module in order to get the most from your system.

How to Use This Documentation

Read the appropriate section of the user guide before proceeding with any system customizations. The user guide presents the procedures and steps required for completing the various Application Server processes. To assist you in locating information, the user guide contains:

- A “Table of Contents” of logically organized activities and tasks
- An alphabetized “Quick Reference Task List” of commonly performed tasks
- An alphabetized “Index” of the information provided in the user guide
Quick Reference Task List
This list contains tasks that are commonly performed with the Application Server module. Each task is cross-referenced to a specific topic.

How Do I Set Up...?
- An Application Server – see “Setting Up an Application Server” on page 6.

How Do I Launch...?
- An Application Server – see “Starting Application Server” on page 14.

How Do I Check...?
- The Application Server Queue – see “Checking the Application Server Queue” on page 15.
- The Application Server Logs – see “Viewing Application Server Logs” on page 17.
- The Information of All Application Servers – see “Viewing Information about All Application Servers” on page 20.

How Do I Change...?
- The Application Server Properties – see “Changing Application Server Properties and Options” on page 22.

How Do I Delete...?
- An Application Server – see “Deleting an Application Server” on page 23.
- Requests from the Application Server Queue – see “Deleting Requests from the Application Server Queue” on page 35.

How Do I Pause...?
- An Application Server – see “Pausing Application Servers” on page 19.

How Do I Restart...?
- An Application Server – see “Restarting Application Servers” on page 19.

How Do I Shut Down...?
- An Application Server – see “Shutting Down Application Servers” on page 19.

How Do I Submit...?
- Requests to the Application Server – see “Submitting Requests” on page 27.
- Send email requests to the Application Server – see “Submitting Requests Via Email” on page 32.
- A document to the Application Server for transmission to a customer, vendor, or employee – see “Processing Requests Created by Quick Send” on page 36.

How Do I Resubmit...?
- Incomplete Requests to the Application Server – see “Resubmitting Incomplete Requests” on page 34.
Administering the Application Server

Overview
The “Administering the Application Server” section provides the information and procedures necessary for administering the Application Server Module (setting up and maintaining Application Servers) in your business. These basic tasks include:

- Setting Up an Application Server
- Starting Application Server
- Checking the Application Server Queue
- Viewing Application Server Logs
-Pausing, Restarting, and Shutting Down Application Servers
- Viewing Information about All Application Servers
- Changing Application Server Properties and Options
- Deleting an Application Server
Setting Up an Application Server

To set up an Application Server, define an Application Server configuration that controls how the Application Server processes requests. The configuration consists of:

- **Application Server name** — Must be the same as a Microsoft Dynamics SL user ID that is defined on User Maintenance (95.260.00).
- **Application Server parameters** — Polling interval (how often to check for new processing requests), email profile (used to receive requests, send responses via email), temporary file directory, and log management settings (when to begin overwriting existing log information with new information).
- **Application Server options** — How the Application Server should handle report routing and requests that did not process successfully (that is, incomplete requests).
- **Application Server groups and users** — The groups and users who can submit requests to the Application Server.

**To set up an Application Server:**

1. Define a user ID and password for the Application Server on User Maintenance (95.260.00). See the System Manager online help or user guide for more information.

   **Note:** This user ID should be set up with access rights to the Application Server (96.010.00) screen and with view rights (at a minimum) to the Application Server Administration (96.080.00) screen.

2. To open Application Server Administration Wizard (96.070.00), click **New Server Wizard** or click **Server Administration**.

![Image of Application Server Administration Wizard](image.png)

*Figure 2: Application Server Administration Wizard (96.070.00)*
3. Click **Next**. The window used to name the new Application Server displays.

![Image of Application Server Administration Wizard](image1.png)

*Figure 3: Naming the new Application Server*

4. Select the name of the Application Server from the list displayed, and then click **Next** to continue.

**Note:** If the Application Server name you want to use has not yet been defined, click **Add User** to access **User Maintenance** (95.260.00). Define the Application Server user name and password, close **User Maintenance** (95.260.00).

The window used to define Application Server polling, maximum requests allowed, email, and temporary directory settings displays.

![Image of Application Server Administration Wizard](image2.png)

*Figure 4: Defining the polling interval, email profile, and temporary directory*
5. Type the number of the polling interval, in seconds, at **Polling Interval** (for example, type 10 to specify that the Application Server should poll the request queue every 10 seconds for new requests).

6. Type the maximum number of requests the Application Server should allow at one time at **Maximum Requests Allowed**.

7. Type the Application Server’s email profile ID at **Email Profile ID**.

**Note:**
- The Application Server email profile defined must be a valid Microsoft Exchange or Outlook profile (for example, the machine where the Application Server will run must have a client installation of either Microsoft Exchange or Outlook and have a profile set up to access email).
- If you plan to utilize the Application Server module to send email requests, be sure that Microsoft Visual Studio Tools for Office (VSTO) add-in is installed on the computer that has been designated as the Application Server. For more information about how to install VSTO, see the Installation Guide.
- The format of the Application Server notification email is optimized for the Courier New font. To make sure that the notification emails line up properly, modify the default font in Outlook. In Outlook, click **Tools**, click **Options**, click **Mail Format**, and then click **Stationery and Fonts**. Under **New mail messages**, click **Font**, and then choose “Courier New.” Click **OK** three times.

8. At **Temporary Directory**, define the directory path and name where the Application Server should store the temporary files it will generate during request processing.

9. Click **Browse** to help locate the directory.

10. Click **Next** to continue. The window used to define Application Server logging options displays.

![Application Server Administration Wizard](image)

**Figure 5: Defining log file maintenance parameters**

11. At **Number of Entries**, type the maximum entries the Application Server should retain in the log before it begins overwriting log information (for example, type 1500 to have the Application Server begin overwriting with the 1501\textsuperscript{st} entry).
12. At **Number of Days**, type the maximum number of days the Application Server should retain information in the log before it begins overwriting log information (for example, type 15 to have the Application Server begin overwriting on day 15).

**Note:** If both entries are set to zero, the log will not be purged automatically and all entries will remain in the log until you purge them manually.

13. Click **Next** to continue. The window used to define when the Application Server should delete completed requests from the request queue displays.

![Application Server Administration Wizard](image.png)

*Figure 6: Defining request queue maintenance parameters*

14. Type the number of days (for example, 10) the Application Server should retain completed request information in the request queue.
15. Click **Next** to continue. The window used to define email report retention options displays.

![Application Server Administration Wizard](image)

**Figure 7:** Selecting the email report retention options

16. Select **Delete non-Quick Send reports after they have been emailed** to delete report files immediately after they are sent via email to their recipients. For historical purposes, email messages containing documents associated with Quick Send requests (purchase orders, invoices, statements, order confirmations, shipping notices, or direct deposit advice slips) are not deleted.

17. Select **Keep the reports in the temporary directory** to maintain the results of a report request in the Application Server temporary file directory even after the reports have been sent via email to the correct recipients.

18. The options under Process Email Requests are specifically for processing requests coming in from an email source. The Application Server will only process email requests that come from email addresses associated with an Active Application Server user. That user has access rights to the **Submit to Application Server** (96.020.00) screen. In the System Manager **User Maintenance** (95.260.00) screen there is a check box that will set a user to be an active Application Server user.

Select one of the following options.

- **OFF, Do not process any emails** — The Application Server will not open the Inbox to look for requests to be processed.

- **Reports** — The Application Server will open the Inbox and look for requests. Only requests that run reports will be processed.

- **All** — The Application Server will open the Inbox and look for requests. All requests will be processed.

19. Select the **Allow email requests to be submitted by SYSADMIN** check box to permit the SYSADMIN user to submit email requests. The Application Server will accept them based on the Process Email Request setting you selected in the previous step. If the check box is **not** selected, email requests by the SYSADMIN user will be ignored. The SYSADMIN is a special case because that user has rights to all the Microsoft Dynamics SL screens.
20. Click **Next** to continue. The window used to define how the Application Server should handle requests that did not finish processing (incomplete requests) displays.

![Image of window for defining request handling](image1.png)

*Figure 8: Selecting a request resubmission option*

21. Select **Automatically resubmit incomplete requests** to have the Application Server automatically resubmit incomplete requests for reprocessing.

22. Select **Leave in queue and flag as “Incomplete”** to have the Application Server assign an Incomplete status to the request and leave it in the request queue. The request can then be resubmitted or deleted from the **Application Server Status (96.030.00)** screen.

23. Click **Next** to continue. The window used to identify the groups and users who can submit a request to the Application Server displays.

![Image of window for identifying submitter groups and users](image2.png)
24. Under Groups, select the check box for each Application Server group.
25. Under Users, select the check box for each Application Server user.

**Note:** Any user will be able to submit requests to the Application Server if no groups and users are selected.

26. Click **Next** to continue. The window that displays next is used to indicate whether the Application Server being defined is the default Application Server for the groups and users identified in steps 24 and 25.

27. Select the check box for each group and user that will use this Application Server as their default Application Server.
28. Click **Next** to continue. The window that displays indicates that the Application Server setup is complete.

![Figure 11: Completing Application Server setup](image)

29. If one or more setup selections need to be changed, click **Back** until arriving at the incorrect selection, make the correct selection, and click **Next** until you return to this window.

30. Click **Finish** to complete the Application Server setup and exit the Application Server Administration Wizard.
Starting Application Server

When you start Application Server, it begins polling the Application Server queue in the database to run any outstanding process and report requests.

To start Application Server:

1. Start Microsoft Dynamics SL. Login (98.000.00) appears. Type the company ID in **Company ID**.
2. Type the Application Server name in **User ID**. The Application Server name must match a defined user ID.
3. Type in your password in **Password**.
4. Click **OK**. The Microsoft Dynamics SL window appears.
5. Open **Application Server (96.010.00)**, which automatically starts the Application Server specified in step 2.

![Figure 12: Application Server (96.010.00)](image)

While an Application Server is running, the **Processing Log** tab displays the current status of request processing operations, including which request is being processed and the status of the request.

A status bar at the bottom of the window shows whether the Application Server is running or stopped and how many outstanding requests are currently waiting in the queue.
Checking the Application Server Queue

You can access the Application Server queue at any time to review the status of any requests currently in the queue. You can display:

- **Current Queue** — Displays all requests that are currently in the queue and are waiting for an Application Server or have already been processed by an Application Server.
- **Master Schedule** — Displays all requests that have scheduling information defined for them.
- **Today’s Schedule** — Displays all requests that have scheduling information defined for them and could process on the current business day.

**To check the Application Server queue:**

1. Click **View Request Queue** to open Application Server Status (96.030.00).

2. In **Display**, select the desired request queue display option:
   - **Current Queue** — Displays all requests that are waiting to run or that have already been run.
   - **Master Schedule** — Displays all requests that have schedules set up.
   - **Today’s Schedule** — Displays all requests that could run on the current business day based on their schedule.

   If you selected Current Queue, click the **Search** button to search for requests that match certain criteria. A small pane displays, allowing you to enter search criteria (request status, user, or description). Only requests matching the criteria are displayed.
3. Select a request from the list that displays, and the click **Request Details**. **Application Server Status (96.03.010)** displays, showing the details of the currently selected processing request.

![Application Server Status (96.03.010), Companies tab](image)

*Figure 14: Application Server Status (96.03.010), Companies tab*

4. Click **Close** to exit the **Application Server Status (96.03.010)** screen and return to **Application Server Status (96.030.00)**.
Viewing Application Server Logs

Each Application Server maintains an activity log which is used to record the processing activity of the Application Server and the processing activity performed on requests in the request queue. At any time, you can review the log information for a specific request or you can view the entire activity log.

Viewing Activity for a Specific Request

The Application Server log contains the processing history of each request:

- Date of processing action
- Time of processing action
- Request ID and description
- Processing action Application Server has performed on request

To view an individual request log:

1. Click View Request Queue to open Application Server Status (96.030.00).
2. In Display, select the desired request queue display option:
   - Current Queue — Displays all requests that are currently in the queue and are waiting for an Application Server or have already been processed by an Application Server.
   - Master Schedule — Displays all requests that have scheduling information defined for them.
   - Today's Schedule — Displays all requests that have scheduling information defined for them and could process on the current business day.
3. Select a request from the list of requests displayed, and click Request Details. Application Server Status (96.03.010) displays, showing the details of the currently selected processing request.
4. Click the Request Log tab. A list of the requests processing history appears in Application Server Status (96.03.010).

Figure 15: Application Server Status (96.03.010), Request Log tab
Viewing the Entire Activity Log

The full version of the Application Server log shows the processing history of all requests in the Application Server request queue. Because of this, it is known as the *master request log*.

**To view the master request log:**

1. Click **Application Server Log** to open Application Server Request Log (96.060.00).

![Figure 16: Application Server Request Log (96.060.00)](image)

2. Select the Application Server name at **Display Request Log For**. The request log information displays in Application Server Request Log (96.060.00).

3. Click the plus sign to the left of a request ID to view request processing history.

**Note:** This does not have to be completed from the machine where the Application Server is located. It can be completed from any machine with access to Application Server Administration (98.080.00).
Pausing, Restarting, and Shutting Down Application Servers

Once an Application Server is launched, you can pause, restart, or shut down the Application Server to match workload requirements.

**Example:** If requests are backing up in the queue, you can launch additional Application Servers to meet the demand. When the request load goes back down, you can pause or stop the idle servers and then start them up again when the workload picks up again.

Whether you pause or stop the Application Server, the requests remain in the queue. If there are other Application Servers running, they pick up and process those requests.

**Pausing Application Servers**

Outstanding requests that have not been retrieved from the queue when you click **Stop** remain in the Application Server queue for later processing.

**To pause an Application Server:**

Click **Stop** on *Application Server (96.010.00)*. Application Server processing pauses.

**Restarting Application Servers**

If you pause an Application Server during the current session, restart the current Application Server so that request processing resumes.

**To restart an Application Server:**

Click **Start** on *Application Server (96.010.00)*. Processing resumes.

**Shutting Down Application Servers**

System Administrators can shut down the current Application Server at any time by closing *Application Server (96.010.00)*. You must launch the Application Server to restart it. See “Starting Application Server” on page 14.

**Using Alternate Methods**

In addition to the pause, restart, and shutdown procedures described earlier in this section, you can also pause, restart, or shut down an Application Server by following these steps:

1. **Accessing Application Server Administration (98.080.00).**
2. Selecting the Application Server and clicking the right mouse button.
3. Selecting the appropriate option from the pop-up menu.

**Note:** This does not have to be completed from the machine where the Application Server is located. It can be completed from any machine with access to *Application Server Administration (96.080.00).*
Viewing Information about All Application Servers

When you are operating multiple Application Servers simultaneously, being able to obtain an overview of the current information (status and activity) of all Application Servers can help you decide how to distribute the request load to improve Application Server processing efficiency. Application Server includes functions that can quickly give you this broad overview of Application Server operations.

To view information about all currently running Application Servers:

1. Click Server Administration to open Application Server Administration (96.080.00).

![Application Server Administration (96.080.00)](image1)

Figure 17: Application Server Administration (96.080.00)

2. On the View menu, click Details. The “view” of Application Server Administration (96.080.00) changes from icon view to details view.

![Application Server Administration (96.080.00) — View by Details](image2)

Figure 18: Application Server Administration (96.080.00) — View by Details
The details of each Application Server include server name, status (shut down, paused, operating), and information about any request currently being processed. Use this information to help determine if part of the overall Application Server request load could be shifted to an Application Server that is currently idle.
Changing Application Server Properties and Options

Each Application Server may perform differently depending on the properties and options defined for the Application Server in Application Server Administration (96.080.00). Each Application Server’s properties and options consist of:

- General settings and options: polling interval, email profile ID, temporary file directory, and report routing and request resubmission options
- Log management (that is, when to overwrite existing processing and request log information with new information)
- Groups and users that can submit requests to the Application Server
- Groups and users for which the Application Server is the default Application Server

An Application Server’s properties and options are defined when you first set up the Application Server (see “Setting Up an Application Server” on page 6). However, you can change an Application Server’s properties and options at any time after Server setup.

To change an Application Server’s properties and options:

1. Click Server Administration to open Application Server Administration (96.080.00).
2. Double-click an Application Server or select an Application Server, and then click Properties. Application Server Properties displays.

Figure 19: Application Server Properties, General tab

3. Click on the correct Application Server Properties tabs, and change the Application Server’s properties and options as necessary. See “Application Server Properties” on page 69 for information about completing the Application Server Properties tabs.
4. Click Apply to apply the new properties and options to the Application Server.
5. Click OK to exit Application Server Properties and return to Application Server Administration (96.080.00).
Deleting an Application Server

If an Application Server is no longer needed for processing, you can delete the Application Server. This removes all record of the Application Server in the database.

**Note:** Before deleting an Application Server, make sure all requests to be processed by the Server have been processed. Otherwise, those requests will not be processed.

**To delete an Application Server:**
1. Click **Server Administration** to open **Application Server Administration** (96.080.00).
2. Select the Application Server to delete, and then click **Delete Server**. **Application Server Administration** asks you to confirm that you want to delete the selected Server.
3. Confirm that you want to delete the Application Server.
Using Application Server

Overview
The “Using Application Server” section provides the information and procedures necessary for submitting requests to the Application Server and managing those requests. These basic tasks include:

- Submitting Requests
- Submitting Requests Via Email
- Resubmitting Incomplete Requests
- Deleting Requests from the Application Server Queue
- Processing Requests Created by Quick Send

Types of Requests
The Application Server carries out several types of requests: process requests, report requests, Quick Send requests, Doc Share requests, and transaction import (TI) requests.

- A process request is an Application Server request originating from a process screen such as Release General Ledger Batches (01.400.00) in the General Ledger module.

- A report request is a request originating from a report screen such as Inventory Valuation (10.620.00) in the Inventory module.

- A Quick Send request is a request created when a purchase order, invoice, statement, order confirmation, shipping notice, or direct deposit advice slip is sent to a vendor, customer, or employee via email or fax using the Quick Send functionality in the Shared Information module. A Quick Send request originates from:
  - Print Purchase Orders (04.600.00) in the Purchasing module
  - Invoice and Memo (08.760.00) in the Accounts Receivable module
  - Statements (08.600.00) in the Accounts Receivable module
  - Invoice (40.680.00) in the Order Management module
  - Order Confirmation (40.610.00) in the Order Management module
  - Manual Order Confirmation (40.611.00) in the Order Management module
  - Shipping Notice (40.654.00) in the Order Management module
  - Invoice Print (BI.INV.00) in the Flexible Billings module
  - Construction Billing Print (BL.CNP.00) in the Flexible Billings module
  - Service Call Invoice (SD.640.00) in the Service Dispatch module
  - Reprint Service Call Invoice (SD.650.00) in the Service Dispatch module
  - Service Contract Invoice (SN.640.00) in the Service Contracts module
  - Reprint Service Contract Invoice (SN.650.00) in the Service Contracts module
  - Direct Deposit Advice Slips (02.635.00) in the Payroll Module

- A Doc Share request is a request that is created to publish a customer, vendor, or project document on a SharePoint site. A Doc Share request can originate from:
  - Invoice and Memo (08.760.00) in the Accounts Receivable module
  - Statements (08.600.00) in the Accounts Receivable module
  - Invoice (40.680.00) in the Order Management module
  - Order Confirmation (40.610.00) in the Order Management module
- Shipping Notice (40.654.00) in the Order Management module
- Print Purchase Orders (04.600.00) in the Purchasing module
- Invoice Print (BI.INV.00) in the Flexible Billings module
- Construction Billing Print (BI.CNP.00) in the Flexible Billings module

- A transaction import (TI) request is a request that imports data from an external program (databases, spreadsheets, etc.) into the database. TI requests originate from Transaction Import (98.500.00), a system-level Microsoft Dynamics SL window.

Note: When a TI request is submitted to Application Server, Application Server reads the TI data file and the TI control macro and includes their contents with the TI request. Once the request has been submitted, the original TI data file and control macro can be deleted.
Submitting Requests

When you submit a request to the Application Server queue, the Application Server module automatically creates template information specific to the submitted process, report, or Transaction Import (TI) request. This template ensures that the request is completed exactly as if it were run directly from a workstation.

Submitting an Application Server request involves defining the request's parameters. These parameters consist of:

- General request options, including which Companies the request will run against.
- Scheduling information (that is, when processing should occur) for standard and recurring requests.
- Routing information for requests that generate reports to be distributed via email.
- The Application Server and template to use for request processing.

You can submit requests to the Application Server queue even if no Application Servers are running. Requests are stored in the queue until at least one Application Server is launched and begins processing the requests. Also, you can run each request against more than one company. After processing a request, the Application Server updates the status of the request in the queue to show that the request is complete.

Note:

- You can also submit requests via email, which is useful for running processes or obtaining reports when you are away from the office and accessing the software directly is not possible or practical. To learn how to send your email Application Server requests to an Application Server, see “Submitting Requests Via Email” on page 32.
- In order to submit requests to an Application Server via email, the Application Server must have a valid Microsoft Exchange- or Outlook-based email profile ID (that is, the machine where the Application Server is running must have a client installation of either Microsoft Exchange or Outlook).
- For security reasons, consider locking the console of the computer you have set up to process Application Server requests. Since the Application Server needs to be running to process most requests, locking its console will help ensure that unwelcome access does not occur. The Application Server console must be unlocked only to process Transaction Import requests.
- The format of the Application Server notification email is optimized for the Courier New font. To make sure that the notification emails line up properly, modify the default font in Outlook. In Outlook, click Tools, click Options, click Mail Format, and then click Stationery and Fonts. Under New mail messages, click Font, and then choose “Courier New.” Click OK three times.

To submit an Application Server request from within Microsoft Dynamics SL:

1. At the workstation, access a process screen or report, or the Transaction Import (98.500.00) window. See the System Manager online help or user guide for more information.
2. Complete the fields and options of the currently open window, but do not click Begin Processing to run the process or Print to print the report.
3. On the **Actions** menu, click **Submit to Application Server**. **Submit to Application Server** (96.020.00) appears, showing the request type and the number of the screen from which you initiated the request.

![Submit to Application Server (96.020.00), General tab](image)

**Figure 20:** Submit to Application Server (96.020.00), General tab

4. Type a description of the request at **Request Description**.

5. Select the request’s processing priority (high, normal, low) at **Priority**.

6. If you want to be notified when Application Server has finished processing your request, select the **Notify me by email when the request has finished processing** check box, and then type your email address in the **Email Address** box.

7. Click the **Selected** check box next to the companies against which you want to run the request.

**Note:** When submitting report requests in a multi-company installation, the Report Option Interpreter (ROI) provides the ability to select which companies a report will be run against. If you select companies within the ROI when submitting a report, then it is not possible to select companies in **Submit to Application Server** (96.020.00). This is because when the report is run by Application Server, it will automatically be run against the companies selected in the ROI.
8. Click the **Scheduling** tab to access request scheduling fields.

![Submit to Application Server (96.020.00), Scheduling tab](image)

*Figure 21: Submit to Application Server (96.020.00), Scheduling tab*

9. Select how often the request should be processed at **Scheduling Frequency**. The option you select determines the **Scheduling** tab fields that display below **Scheduling Frequency**.

10. Complete the request scheduling fields and options as necessary. See “Submit to Application Server (96.020.00)” on page 43 for information on completing the fields associated with the scheduling frequency option selected.

**Note:** The Daily, Weekly, Monthly, and Event scheduling options create recurring Application Server requests. These types of requests will automatically be run by Application Server each time their associated request parameters are met (for example, if a monthly request is set up to process on the third Monday of each month, Application Server will run the request automatically on the third Monday of each month).
11. Click the **Report Routing** tab to access fields used for routing the results of report requests via email.

**Note:** Skip this step if the request is not a report request or if it is a report request that will not be routed through email.

![Figure 22: Submit to Application Server (96.020.00), Report Routing tab](image)

12. In **Report Format**, select the file format in which the report is to be submitted to its recipients.
13. In **Email Text**, type an explanatory email message to accompany the report.
14. In **User ID**, type the recipient’s user ID.
15. In **Email Address**, type the email address you would like to use. It can be any valid email address and does not have to be that of a user.

**Note:** If the request is a recurring request (its **Scheduling Frequency** option is Daily, Weekly, Monthly, or Event), the report routing information becomes the default routing information for the recurring request.
16. Click the **Advanced** tab to access fields used for selecting the Application Server to process the request and to attach template specifications to the request.

**Note:** Skip this step if the request can be processed by the first available Application Server and uses the default processing template option **Template as the active application** (use the settings of the application where the request originated as the processing template).

![Submit to Application Server (96.020.00), Advanced tab](image)

17. In the **Submit To** box, select the Application Server that should process the request (that is, the Application Server to which the request should be submitted).

18. Under Transaction Import Options, select the import option to use with the data file and control macro.

19. Under **Templating**, select the template that should be used to process the request. The processing template can be the current template of the application originating the request, the default settings of the application (that is, no template), or another named application template.

**Note:** If the request is a recurring request (its **Scheduling Frequency** option is Daily, Weekly, Monthly, or Event), the Application Server and templating information defined in the **Advanced** tab becomes the default Application Server and template for the recurring request.

20. Click **Submit** to send the request to the specified Application Server request queue.

21. Click **Close** to exit the **Submit to Application Server (96.020.00)** screen without submitting a request.
Submitting Requests Via Email

When you are away from the office and access is not possible or practical, you can still run processes and obtain reports by submitting your application processing and reporting requests to an Application Server via an email connection.

Note: The Application Server will only process email messages that have been submitted from an email address that is associated with a user. The user must be an active Application Server user and have access rights to the Submit to Application Server (96.020.00) screen. There are several special settings available for email processing when setting up an Application Server.

To submit an Application Server request via email:

1. Establish an Internet connection and start your Web browser.
2. Access the browser's email feature to create the Application Server email request.
3. In the **Subject** box, type Request (or whatever information identifies the subject of the email message).
4. In the body of the email request message, define the Application Server request, using the following fields and observing the following request submission rules:

   ```
   Userid:
   Screen:
   Description:
   Priority:
   Template ID:
   Parameters:
   Company:
   [Distribution]
   Format:
   Notify:
   Email:
   [Options]
   Logging:
   Errors:
   Processing:
   [Screen]
   Number:
   Data:
   Control:
   Output:
   ```

   a) **Entries in Userid:** and **Screen:** are required.
   b) **Use Company:** to specify the company (companies) against which the request should be run.
   c) **Format:** specifies the file format in which Application Server should save a request generated report for distribution. Valid formats are WORD (Microsoft Word), HTML (HTML 3), EXCEL (Microsoft Excel), TEXT (text), CSV (comma separated value), WORDPERFECT (WordPerfect), LOTUS (Lotus 1-2-3), CRYSTAL (Crystal Reports), RICHTEXT ( Rich Text), PDF ( Acrobat PDF ),XML ( Extended Markup Language ).
   d) **Use Notify:** to define the email locations to be notified when the email request is finished processing (for example, Notify: bob@acme.com). Email addresses must be separated by semicolons (for example, Notify: bob@acme.com; tim@acme.com). You can also type No if you do not want to send notification to any email addresses (for example, Notify: No).
e) Use **Email**: to define the email locations where the report generated by a report request is to be distributed (for example, Email: bob@acme.com). Email addresses must be separated by semicolons (for example, Email: bob@acme.com; tim@acme.com).

f) Define the [Options] section only when submitting Transaction Import (TI) requests. Use **Logging**: to define logging options (All, None, or Error). Use **Errors**: to define how many processing errors TI can encounter before it stops processing. Use **Processing**: to define the TI processing option (Edit, Combined, or Update).

g) Define the [Screens] section only when submitting TI requests (add one [Screen] section for each screen TI is to run). Use **Number**: to define the screen number that TI should run (for example, 01.010.00). Use **Data**: to define the data file (attached to the email message) to run against the screen defined at **Number**. Use **Control**: to define the control file (attached to the email message) that is to control the screen defined at **Number**. Use **Output**: to define the name of the output file TI will create.

**Note**: You can obtain the Application Server email request submission template (a predefined template containing the fields **Userid**: through **Output**) by sending an email message to Application Server requesting the template. Use the subject Command and the message body Template in your email request. After receiving your request, Application Server will send you the template.

![Figure 24: Application Server Email Submission Request Template](image)

5. Submit the Application Server request via email to the specified Application Server. Application Server will process the message and return an email response, indicating whether or not the request was successfully submitted to the Application Server queue.
Resubmitting Incomplete Requests

Occasionally, an Application Server request may not process successfully. The Application Server module identifies any unsuccessfully processed request as an “incomplete” request. You can attempt to process incomplete requests again by resubmitting them to the Application Server.

To resubmit an incomplete Application Server request:

1. Click View Request Queue to open Application Server Status (96.030.00).
2. In Display, select the Current Queue display option and review the request list for incomplete-status requests.
3. Select the incomplete request in Application Server Status (96.030.00).
4. Click Resubmit Request to resubmit the incomplete request to the request queue. The Application Server will attempt to process the request again.
Deleting Requests from the Application Server Queue

After running a process or report request, the Application Server updates the status of the request in the queue to show that the request is complete. To keep the list in the queue at a manageable level, delete requests that are completed. You may also need to delete requests from the queue for maintenance and troubleshooting. Only requests that have a status of complete, waiting, or incomplete can be deleted.

The System Administrator can delete any request. Users can delete only the requests they submitted.

To delete a request from the Application Server queue:

1. Click View Request Queue to open Application Server Status (96.030.00).
2. In Display, select the desired request queue display option and review the request list requests you wish to delete.
3. Click to select the Delete check box next to each request to delete.
4. Close Application Server Status (96.030.00). The request(s) marked for deletion are removed from the Application Server queue.
Processing Requests Created by Quick Send

Quick Send requests are created and automatically submitted to Application Server when documents for vendors, customers, or employees that are set up for Quick Send are processed during a print routine. You can transmit the following documents electronically using Quick Send and Application Server:

- **Print Purchase Orders** (04.600.00) in the Purchasing module
- **Invoice and Memo** (08.760.00) in the Accounts Receivable module
- **Statements** (08.600.00) in the Accounts Receivable module
- **Invoice** (40.680.00) in the Order Management module
- **Order Confirmation** (40.610.00) in the Order Management module
- **Manual Order Confirmation** (40.611.00) in the Order Management module
- **Shipping Notice** (40.654.00) in the Order Management module
- **Invoice Print** (BI.INV.00) in the Flexible Billings module
- **Construction Billing Print** (BI.CNP.00) in the Flexible Billings module
- **Service Call Invoice** (SD.640.00) in the Service Dispatch module
- **Reprint Service Call Invoice** (SN.640.00) in the Service Contracts module
- **Service Contract Invoice** (SN.650.00) in the Service Contracts module
- **Direct Deposit Advice Slips** (02.635.00) in the Payroll Module

Application Server polls the queue and processes Quick Send requests like report requests. However, a document created for a Quick Send request is attached to an email message or fax that is transmitted to the recipient according to their Quick Send preferences. For more information about implementing Quick Send, see “Setting up Quick Send” in the Shared Information online help or user guide.

**Note:**

- If Application Server is not registered, all screens in the Application Server module can be accessed except **Submit to Application Server** (96.020.00) because Quick Send requests are automatically submitted to Application Server.
- All Application Server requests can be viewed in the Application Server screens. To view Quick Send requests alone, use **Quick Send Inquiry** (21.200.00) in the Shared Information module.
- In order to send Quick Send requests via email, the Application Server must have a valid Microsoft Exchange- or Outlook-based email profile ID (that is, the machine where the Application Server is running must have a client installation of either Microsoft Exchange or Outlook).
- To send Quick Send requests via fax, Fax Services must be loaded on the computer running Application Server as well as the computer hosting the fax machine. See “Using Fax” in Windows online help to set up Fax Services.
- A fax cover page can be included in a Quick Send request. The fax cover page template, QSCoverPage.cov, is located in \Program Files\Microsoft Dynamics\SL\Applications\AS and can be modified using the Windows utility, Fax Cover Page Editor. If the template is modified, save the file in \Program Files\Microsoft Dynamics\SL\Applications\Usr_Rpts. Information from **Company Setup** (13.250.00) in the Multi-Company module, **Quick Send Setup** (21.951.00) in the Shared Information module, **Vendor Maintenance** (03.270.00) in the Accounts Payable module, **Customer Maintenance** (08.260.00) in the Accounts Receivable module, and **Employee Maintenance** (02.250.00) in the Payroll module is used to prepare the fax cover page.
- For security reasons, consider locking the console of the computer you have set up to process Application Server requests. Since the Application Server needs to be running to process all Quick Send requests, locking its console will help ensure that unwelcome access does not occur.
Processing Requests Created by Doc Share

You can create Doc Share requests that are submitted to Application Server to post a customer, vendor, or project document on a SharePoint site. Requests are generated based on Doc Share configuration settings in the System Manager and Shared Information modules. Application Server polls the queue for Doc Share requests and processes them in a way that is similar to how it handles report requests. However, it transmits a document created from a Doc Share request to a SharePoint site instead of sending it to a printer.

You can create Doc Share requests in

- Invoice and Memo (08.760.00) in the Accounts Receivable module
- Statements (08.600.00) in the Accounts Receivable module
- Invoice (40.680.00) in the Order Management module
- Order Confirmation (40.610.00) in the Order Management module
- Shipping Notice (40.654.00) in the Order Management module
- Print Purchase Orders (04.600.00) in the Purchasing module
- Invoice Print (BI.INV.00) in the Flexible Billings module
- Construction Billing Print (BI.CNP.00) in the Flexible Billings module

After a customer, vendor, or project document is posted on a SharePoint site by Doc Share and Application Server, you can view it from

- Project Controller Project Maintenance (PA.PRJ.00), Project Net Profit (PA.PNR.00), and Task Net Profit (PA.PND.00)
- Analyzer Project Analyzer (IQ.PAS.00)
- Accounts Receivable Customer Maintenance (08.260.00), Customer History (08.261.00), and Customer Inquiry (08.200.00)
- Accounts Payable Vendor Maintenance (03.270.00), Vendor History (03.271.00), and Vendor Inquiry (03.200.00)

For more information about implementing Doc Share, see the System Manager help or user guide. To find additional information about using Doc Share to publish documents to a SharePoint site, see the user guide or help for the appropriate module.

Note:

- You do not need to purchase a license for Application Server in order to configure Doc Share for use with an application. All Application Server screens, except Submit to Application Server (96.020.00), can be accessed if the module is not registered. However, you will not need this screen for your Doc Share requests since they are submitted automatically.
- For security reasons, consider locking the console of the computer you have set up to process Application Server requests. Since Application Server needs to be running to process Doc Share requests, locking its console will help ensure that unwelcome access does not occur.
Reference

Application Server Module Menu

Used to access the screens and reports in the Application Server module. See the System Manager online help or user guide for more information on using module menus.

Figure 27: Application Server Module menu
**Application Server (96.010.00)**

Once started, an Application Server begins polling the Application Server queue in the database for outstanding process, report, Quick Send, and Doc Share requests contained in the queue. The Application Server runs the requests according to user-specified and templated information contained in each request.

![Application Server (96.010.00)](image)

**Start (button)**

Restarts the current Application Server so that request processing resumes.

**Stop (button)**

Pauses the current Application Server processing; outstanding requests remain in the Application Server queue for later processing.

**Running (message)**

Shows whether selected Application Servers are *Running* or *Paused*.

**Outstanding Requests (message)**

Shows the number of requests in the Application Server queue.
Application Server, Processing Log Tab

Used to view the contents of the Application Server processing log maintained in the database. The Application Server processing log includes:

- Application Server starts, stops, pauses.
- User request and request completed data.
- Email requests, responses, and “process complete” notifications.
- Deleted user requests.
- Printer used to print report requests.
- Incomplete request management data (for example, how Application Server handled incomplete user requests).

Figure 29: Application Server (96.010.00), Processing Log tab
Application Server, Queue Status Tab

Used to view the outstanding requests in the Application Server request queue.

![Image of Queue Status Tab](image)

**Figure 30: Application Server (96.010.00), Queue Status tab**

**ID**
Identification code of the Application Server request.

**User ID**
Identification code of the user submitting the request.

**Description**
Brief, user-defined description associated with the request.

**Screen**
Number of the screen where the request originated (for example, 10620, which represents a report request originating from *Inventory Valuation* (10.620.00) in the Inventory module).

**Priority**
The relative importance of the request being submitted:

- **High** — The request has high priority in the Application Server queue; automatically runs before normal- and low-priority requests.
- **Normal** — The request has average priority; automatically runs after high-priority requests and before low-priority requests.
- **Low** — The request has low priority; automatically runs after high- and normal-priority requests.
Submit to Application Server (96.020.00)

Used to define the parameters of each process, report, and transaction import (TI) request submitted to the Application Server request queue. These parameters consist of:

- General request options, including which companies the request will be run against.
- Scheduling information (when processing should occur) for standard and recurring requests.
- Routing information for requests that generate reports to be distributed via email.
- The Application Server and template option to use for request processing.

![Submit to Application Server (96.020.00), General tab](image)

**Figure 31: Submit to Application Server (96.020.00), General tab**

**Request Screen**
Number of the screen where the process or report request originated, such as 10.620.00, which represents an inventory valuation report request originating from Inventory Valuation (10.620.00) in the Inventory module.

**Request Type**
Specifies the type of request, either process or report, being submitted.

**Request Description**
Brief, user-defined description associated with the request being submitted.

**Priority**
The relative importance of the request being submitted. Options are:

- **High** — The request has high priority in the Application Server queue; automatically runs before normal- and low-priority requests.
• Normal — The request has average priority; automatically runs after high-priority requests and before low-priority requests.
• Low — The request has low priority; automatically runs after high- and normal-priority requests.

Submit (button)
Saves the current request to the Application Server queue. The Application Server module then automatically creates template information specific to the process or report submitted, ensuring that the request is completed exactly as if it were run at the workstation.

Close (button)
Close the screen without submitting a request to the Application Server queue.
Submit to Application Server, General Tab

Used to specify whether or not Application Server should provide email notification when it has completed the current request and to identify the company (companies) against which Application Server should run the request.

The Notification options are used to define whether or not Application Server should provide email notification when the request is complete and to identify the email addresses where notification should be sent.

The Company Selection options are used to select the company or companies against which the current Application Server request should be run.

![Submit to Application Server (96.020.00), General tab](image)

**Figure 32: Submit to Application Server (96.020.00), General tab**

**Notify me by email when the request has finished processing** (check box)
Select to have Application Server provide email notification when the request is complete.

**Email Address**
The email addresses where Application Server should send request complete notification. Multiple email addresses can be specified by separating the addresses with semicolons.

**Selected** (check box)
Specifies whether or not the request should be run against the company listed. One or more companies can be selected for each process or report request. Allows for running a request against more than one company or against a company other than the default company from which the request was submitted.

**Company Name**
Company to run the current Application Server request against, if selected.
Submit to Application Server, Scheduling Tab

Used to define when the current Application Server request should be processed: date, time, day of week, day in month, etc.

![Figure 33: Submit to Application Server (96.020.00), Scheduling tab](image)

**Scheduling Frequency**

How often the Application Server request should be processed. Options are:

- **No Scheduling** — The request should be processed when it is the oldest No Scheduling request in the queue; requests of this type are processed on a FIFO basis.
- **One Time** — The request should be processed only once, at a specific date and time.
- **Daily** — The request should be processed daily.
- **Weekly** — The request should be processed weekly.
- **Monthly** — The request should be processed monthly.
- **Event** — The request should be processed after another Application Server request(s) has been processed.
No Scheduling

Used to specify that the current request should be processed when it is the oldest request in the Application Server queue, based on a FIFO request scheduling scheme (that is, when its turn comes up in the Application Server request queue).

![Submit to Application Server (96.020.00), Scheduling tab — No Scheduling](image)

Figure 34: Submit to Application Server (96.020.00), Scheduling tab — No Scheduling
One Time Scheduling Frequency

Used to specify that the request should be processed one time on a certain date and to define the time and frequency of processing.

![Submit to Application Server (96.020.00), Scheduling tab — One Time frequency](image)

**Start Time**

Specifies the hour, minute, and time of day (a.m. or p.m.) that the request should be processed.

**Time Range**

Specifies a range of time and frequency that the request should be processed on any particular business day (12:00 AM to 11:59 PM). A start time and end time can be set, as well as a frequency based on hours and minutes. The end time must be later in the day than the start time. The frequency settings must be five minutes or longer.

**Example:** These settings will allow you to submit a request to process from 1:00 AM to 6:00 AM every 15 minutes. However, you cannot submit a request to process from 10:00 PM to 6:00 AM every 15 minutes. You would need to create two submissions to accomplish this.

**Date To Execute**

Specifies the day, month, and year when the request should be processed.
Daily Scheduling Frequency

Used to specify that the request should be processed on a daily basis and to define the time and frequency of processing.

![Figure 36: Submit to Application Server (96.020.00), Scheduling tab — Daily frequency](image)

**Start Time**

Specifies the hour, minute and time of day (a.m. or p.m.) when the request should be processed.

**Time Range**

Specifies a range of time and frequency that the request should be processed on any particular business day (12:00 AM to 11:59 PM). A start time and end time can be set, as well as frequency based on hours and minutes. The end time must be later in the day than the start time. The frequency settings must be five minutes or longer.

**Example:** These settings will allow you to submit a request to process from 1:00 AM to 6:00 AM every 15 minutes. However, you cannot submit a request to process from 10:00 PM to 6:00 AM every 15 minutes. You would need to create two submissions to accomplish this.

**How Often**

The daily frequency to use for processing the request. The options are:

- **Every Day** — The request should be processed daily, seven days a week.
- **Weekdays Only** — The request should be processed daily, Monday through Friday.
- **Every ___ Days** — The request should be processed every other day, every third day, every fifth day, etc.
Weekly Scheduling Frequency

Used to specify that the request should be processed on a weekly basis and to define the time and frequency of processing.

![Submit to Application Server (96.020.00), Scheduling tab — Weekly frequency](image)

**Figure 37: Submit to Application Server (96.020.00), Scheduling tab — Weekly frequency**

**Start Time**

Specifies the hour, minute, and time of day (a.m. or p.m.) when the request should be processed.

**Time Range**

Specifies a range of time and frequency that the request should be processed on any particular business day (12:00 AM to 11:59 PM). A start time and end time can be set, as well as frequency based on hours and minutes. The end time must be later in the day than the start time. The frequency settings must be five minutes or longer.

**Example:** These settings will allow you to submit a request to process from 1:00 AM to 6:00 AM every 15 minutes. However, you cannot submit a request to process from 10:00 PM to 6:00 AM every 15 minutes. You would need to create two submissions to accomplish this.

**Every __ week(s)**

Identifies how often (every week, every other week, every third week, etc.) the request should be processed.

**Day of Week (check boxes)**

The days of the week on which the request should be processed. The request is processed on the days selected only during the week identified at Every__ week(s).
Monthly Scheduling Frequency

Used to specify that the request should be processed on a monthly basis and to define the time and frequency of processing.

![Figure 38: Submit to Application Server (96.020.00), Scheduling tab — Monthly frequency](image)

**Start Time**

Specifies the hour, minute, and time of day (a.m. or p.m.) when the request should be processed.

**Time Range**

Specifies a range of time and frequency that the request should be processed on any particular business day (12:00 AM to 11:59 PM). A start time and end time can be set, as well as frequency based on hours and minutes. The end time must be later in the day than the start time. The frequency settings must be five minutes or longer.

**Example:** These settings will allow you to submit a request to process from 1:00 AM to 6:00 AM every 15 minutes. However, you cannot submit a request to process from 10:00 PM to 6:00 AM every 15 minutes. You would need to create two submissions to accomplish this.

**Day (Monthly Execution option)**

The numeric day of the month when the request should be processed: first day of the month, 15 day of the month, etc.

**The (Monthly Execution option)**

The named day of the month when the request should be processed: first Monday of the month, second Friday of the month, etc.

**Of the following months (Monthly Execution option)**

The month(s) of the year in which the request should be processed.
Event-Based Scheduling

Used to specify that the request should be processed only after another Application Server request(s) has been processed.

![Image of Event-Based Scheduling](image)

Figure 39: Submit to Application Server (96.020.00), Scheduling tab — Event-based Scheduling

Start Time

Specifies the hour, minute and time of day (a.m. or p.m.) when the request should be processed.

Scheduled Requests

Currently schedule Application Server requests that can potentially be required to be processed before the current request can be processed.

Right-arrow button

Click to move an Application Server request name from **Scheduled Requests** to **Selected Requests**. This identifies that the moved request must be processed before the current Application Server request can be processed.

Left-arrow button

Click to move an Application Server request name from **Selected Requests** to **Scheduled Requests**. This identifies that the moved request does not have to be processed before the current Application Server request can be processed.

Selected Requests

An Application Server request(s) that must be processed before the current request can be processed.
Submit to Application Server, Report Routing Tab

Used to identify which users should receive email copies of reports after an Application Server report request has finished processing.

![Submit to Application Server (96.020.00), Report Routing tab](image)

**Figure 40: Submit to Application Server (96.020.00), Report Routing tab**

**Report Format**
The file format in which the report should be submitted to its recipient.

**Email Text**
The text of the email message accompanying the report.

**Recipients**
The user ID(s) and email address(es) of the user(s) to receive the report.
Submit to Application Server, Advanced Tab

Used to select the Application Server that will process the request and to select the template option to use for request processing: no template (default application settings), default application template, another named template.

**Figure 41**: Submit to Application Server (96.020.00), Advanced tab

**Destination Application Server**

The Application Server to process the current request. Selecting Any specifies that the request should be processed by the first available Application Server.

**Transaction Import Options**

Specify how Application Server uses the Data File and Control Macro. Options are:

- Save the contents of the files with the request at the time it is submitted.
- Use the files directly from the specified location when the request is run.

**Templating**

The template Application Server should use to process the request. Options are:

- **Template the active application** — Process the request using the current application as the request template.
- **No template. Application will run in default state** — Process the request using the application in its default state (no template is applied).
- **Apply selected template** — Process the request using a selected template. Note that this option is disabled if there are no templates for the current screen in the database.
Application Server Status (96.030.00)

Used to review and manage the status of process, report, Quick Send, and Doc Share requests currently in the Application Server queue. Includes display and request selection tools that can narrow the review focus. Review the status of:

- The Current Application Server queue.
- The Master Application Server schedule.
- Today’s scheduled Application Server processing requests.

![Application Server Status (96.030.00)](image)

**Figure 42: Application Server Status (96.030.00)**

**Display**

The Application Server request queue information to be displayed. Options are:

- **Current Queue** — Displays all requests that are currently in the queue and are waiting for an Application Server or have already been processed by an Application Server.
- **Master Schedule** — Displays all requests that have scheduling information defined for them.
- **Today’s Schedule** — Displays all requests that have scheduling information defined for them and could process on the current business day.

**Search (button)**

Available when the Current Queue is displayed. Clicking **Search** accesses a dialog used to specify search criteria — request status, user ID, and/or description — that can be used to filter the queue display.
Grid/Form (when Current Queue option is selected)

This is how the screen looks when the Current Queue option is selected in Display. This option is used to display all requests that are currently in the queue and are waiting for an Application Server or have already been processed by an Application Server.

![Image of Application Server Status]

**Figure 43: Current Queue Display of Application Server Status (96.030.00)**

**ID**
Numerical request ID associated with a specific process or report request in the Application Server queue; indicates the position of the request in the queue.

**Delete**
Specifies whether or not a request should be deleted from the Application Server queue.

**User ID**
User who submitted the request to the Application Server queue.

**Status**
Current status of the request in the Application Server queue. Options are:

- **Waiting in Queue** — Requests submitted to the Application Server queue but waiting to be processed.
- **Processing** — Requests that have been retrieved from the queue by the Application Server but are not yet executing.
- **Executing** — Requests being processed by the Application Server.
- **Complete** — Requests that are completed.
- **Incomplete** — Requests that did not process successfully; these can be resubmitted for processing.
- **Security** — Requests that were processed but failed to run because the user did not have rights to access the requested screen or the specified user ID is no longer valid in the software.
• Hold — Requests submitted to the Application Server but not yet added to the queue.

**Description**
User-defined description associated with the request being submitted, as defined on *Submit to Application Server (96.020.00)*.

**Screen**
Number of the screen where the process or report request originated.

**Type**
Request type, either *process* or *report*, as defined on *Submit to Application Server (96.020.00)*.

**Priority**
Indicates the processing priority of the request in the Application Server queue: high, normal, or low.
**Grid/Form (when Master Schedule option is selected)**

This is how the screen looks when the Master Schedule option is selected for **Display**. This option is used to display all requests that have scheduling information defined for them.

![Image of Grid/Form](image)

**Figure 44: Master Schedule Display of Application Server Status (96.030.00)**

**ID**

Numerical request ID associated with a specific process or report request in the Application Server queue; indicates the position of the request in the queue.

**Delete**

Specifies whether or not a request should be deleted from the Application Server queue.

**User ID**

User who submitted the request to the Application Server queue.

**Description**

User-defined description associated with the request being submitted, as defined on **Submit to Application Server** (96.020.00).

**Screen**

Number of the screen where the process or report request originated.

**Type**

Request type, either **process** or **report**, as defined on **Submit to Application Server** (96.020.00).

**Scheduling Information**

Request scheduling details; includes:
- **Status** (active or inactive).
- **Interval** (one time, daily, weekly, monthly, event).
- **Time** (hour, minute, and time of day when processing is to start).
- **Date** (day, year, and month when processing is to occur).
Grid/Form (when Today’s Schedule option is selected)

This is how the screen looks when the Today’s Schedule option is selected for Display. This option is used to display all requests that have scheduling information defined for them and could process on the current business day.

Figure 45: Today’s Schedule Display of Application Server Status (96.030.00)

ID
Numerical request ID associated with a specific process or report request in the Application Server queue; indicates the position of the request in the queue.

User ID
User who submitted the request to the Application Server queue.

Description
User-defined description associated with the request being submitted, as defined on Submit to Application Server (96.020.00).

Screen
Number of the screen where the process or report request originated.

Type
Request type, either process or report, as defined on Submit to Application Server (96.020.00).

Scheduling Information
Request scheduling details; includes:

- **Scheduling Status** (Waiting or Late).
- **Schedule Basis** (Time, Event).
- **Time** (hour, minute, and time of day when processing is to start).

**Request Details (button)**
Click to access *Application Server Status* (96.03.010), which displays additional information about the currently selected Application Server request.

**Resubmit Request (button)**
Click to resubmit for processing an incomplete-status request.
Application Server Status (96.03.010)

Used to display additional information about the Application Server processing request currently selected in Application Server Status (96.030.00). This display includes:

- The companies to run the request against.
- The name of the event log created when Application Server runs certain process requests. Some processes automatically generate event logs while others do not.

![Application Server Status (96.03.010)](image)

**Figure 46: Application Server Status (96.03.010)**

**Request ID**

Numerical request ID associated with a specific process or report request in the Application Server queue; indicates the position of the request in the queue.

**Request Description**

User-defined description associated with the request being submitted, as defined on Submit to Application Server (96.020.00).

**Close (button)**

Click to close Application Server Status (96.03.010) and return to Application Server Status (96.030.00).
Application Server Status, Companies Tab

Used to view the name of the company (companies) against which the current request is to run, the processing status of the request, and the name of the event log created by the request.

![Application Server Status (96.03.010), Companies tab](image)

Company Name
Company to run the current Application Server request against.

Status
Current status of the request in the Application Server queue: Waiting in Queue, Processing, Executing, Complete, Incomplete.

Event Log File
The name of the event log, if any, created by the software when Application Server runs the request.
**Application Server Status, Request Log Tab**

Used to view the Application Server log information for the selected request:

- Date of processing action
- Time of processing action
- Request ID and description
- Processing action Application Server has performed on request

*Figure 48: Application Server Status (96.03.010), Request Log tab*
**Application Server Status, Schedule Information Tab**

Used to view the processing schedule information of the currently selected Application Server request.

![Application Server Status (96.03.010), Schedule Information tab](Figure 49)

**Scheduling Interval**

How often the request is to be processed: one time, daily, weekly, monthly, event based.

**Scheduling Details**

When the currently selected request is scheduled to begin processing: time of day, day of week, day of month, etc.
Application Server Request Log (96.060.00)

Used to view the Application Server request log processing history for all Application Server requests maintained for a specific Application Server. Request log information includes:

- Date of processing
- Time of processing action
- Request ID and description
- Processing action Application Server has performed on request

Display Request Log For

The Application Server whose request log records should be displayed in Application Server Request Log (96.060.00).

- Selecting All displays the current request logs of all Application Servers.
- Selecting Archive File allows you to open an external file containing historic request log information (that is, request log records at a previous point in time that have been saved to an external file).

Purge Log (button)

Click to delete the records of the currently selected Application Server request log.

Save Log (button)

Click to save the records of the currently selected Application Server request log to an external file.
Application Server Administration Wizard (96.070.00)

Used to create a new Application Server. Open Application Server Administration Wizard (96.070.00) by clicking Add Server in Application Server Administration (96.080.00). The Application Server Administration Wizard (96.070.00) guides you through creating a new Application Server.

Figure 51: Application Server Administration Wizard (96.070.00)

Note: See “Setting Up an Application Server” on page 6 for more information on how to complete the Application Server Administration wizard.
Application Server Administration (96.080.00)

Used to display and change the properties of all currently defined Application Servers and to control their operation (start, stop, pause Application Servers). Also used to delete currently defined Application Servers and to add (define) new Application Servers.

![Application Server Administration (96.080.00)](image)

**Figure 52: Application Server Administration (96.080.00)**

**Note:** To view all currently defined Application Servers by detail rather than by icon, choose View | Details on Application Server Administration (96.080.00). Application Server detail includes Server name, status (shut down, paused, operating) and information about the request currently being processed.

**Add Server (button)**

Click to open Application Server Administration Wizard (96.070.00) in order to create a new Application Server.

**Delete Server (button)**

Click to delete the currently selected Application Server (for example, the one selected in Application Server Administration (96.080.00)).

**Note:** Before deleting an Application Server, make sure all requests to be processed by the Server have been processed. Otherwise, you will have to resubmit them to other Application Servers.

**Properties (button)**

Click to access Application Server Properties, to view and change the properties of the currently selected Application Server.
Application Server Properties

Used to change the property settings of the currently selected Application Server. Application Server properties consist of:

- General Application Server settings and options.
- Log management (that is, when to overwrite existing log information).
- Groups and users that can submit requests to the Application Server.
- Groups and users for which an Application Server is the default Application Server.

![Application Server Properties](image)

*Figure 53: Application Server Properties*
Application Server Properties, General Tab

Used to maintain the general settings and options of the currently selected Application Server: polling interval, email profile, temporary directory, and report routing and request resubmission options.

![Application Server Properties, General tab](image)

Polling Interval
Specifies how often, in seconds, the Application Server should poll the Application Server queue for new Application Server requests to run.

Maximum Requests Allowed
The maximum number of applications that Application Server can run concurrently.

Email Profile ID
The email profile identification code the Application Server uses to access email for receiving processing requests and sending processing results.

Note: Each Application Server email profile defined must be a valid Microsoft Exchange or a Outlook profile (that is, the machine where the Application Server will run must have a client installation of either Microsoft Exchange or Outlook).

Temporary Directory
The directory path and name where the Application Server should store temporary files created during request processing.

Options
How the Application Server should handle completed report routings and incomplete requests.

- Select **Delete non-Quick Send reports after routing** to have the Application Server automatically delete completed report requests after they have been routed to the designated users.
• Select **Automatically resubmit incomplete requests** to have the Application Server automatically resubmit any unsuccessfully processed requests to its request queue.

**Process Email Requests**

These settings are specifically for processing requests coming in from an email source. Application Server will only process requests that come from email addresses associated with active Application Server users. A user must have access rights to the Submit to Application Server (96.020.00) screen. You must also indicate that the user is an Application Server User by selecting the **Active Application Server User** check box in the System Manager User Maintenance (95.260.00) screen.

The **Process Email Requests** options are:

• **OFF, Do not process any emails** — The Application server will not open the Inbox to look for requests to be processed

• **Reports** — The Application server will open the Inbox and look for requests. Only requests that run reports will be processed

• **All** — The Application server will open the Inbox and look for requests. All requests will be processed

**Allow SYSADMIN Email Requests (check box)**

Select this check box to allow the SYSADMIN user to submit requests via email and to accept them based on **Process Email Request** settings. If this check box is **not** selected, emails for the SYSADMIN user will be ignored. The SYSADMIN is a special case because that user has rights to **all** Microsoft Dynamics SL screens.
Application Server Properties, Management Tab

Used to define at what point the Application Server should begin overwriting existing information in its processing and request logs with the information of new processing activity and Application Server requests.

![Application Server Properties, Management tab](image)

**Figure 55: Application Server Properties, Management tab**

**Log Management**

Defines the point at which the Application Server begins to overwrite existing processing log information with new request processing information.

- To have the Application Server begin overwriting existing information after a specific number of log entries, type the number of entries at *Maximum number of entries this Application Server will maintain in the log before overwriting entries* (for example, type 1500 to have the Application Server begin overwriting with the 1501st entry).

- To have the Application Server begin overwriting existing information after a specific number of days, type the number at *Number of days of log data this Application Server will maintain before overwriting entries* (for example, type 14 to have the Application Server begin overwriting on day 15).

**Request Management**

Defines the number of days the Application Server will retain completed requests in the request queue before deleting them automatically.
Application Server Properties, User Assignments Tab

Used to define groups and users that can potentially submit requests to the Application Server.

![Application Server Properties, User Assignments tab](image)

**Groups**
List of groups that can submit requests to the Application Server.

**Add Group (button)**
Click to add a new group to the list of Application Server groups defined at **Groups**. Clicking **Add Group** opens a dialog used to select the new group name.

**Delete Group (button)**
Click to delete an existing Application Server group. To delete a group, select its name at **Groups** then click **Delete Group**.

**Users**
List of users that can submit requests to the Application Server.

**Add User (button)**
Click to add a new user to the list of Application Server users defined at **Users**. Clicking **Add User** opens a dialog used to select the new user name.

**Delete User (button)**
Click to delete an existing Application Server user. To delete a user, select the user name at **Users** then click **Delete User**.

**Note:** If the groups and users sections are not complete (no groups and users defined), any Microsoft Dynamics SL user will be able to submit requests to the Application Server.
Application Server Properties, Default Server Tab

Used to define groups and users for which the Application Server is the default Application Server.

![Application Server Properties - SYSADMIN](image)

**Groups**
List of groups for which the Application Server is the default Application Server.

**Add Group (button)**
Click to add a new group to the list of Application Server groups. Clicking **Add Group** opens a dialog used to select the group name.

**Delete Group (button)**
Click to delete an existing Application Server group. To delete a group, select its name at **Groups** then click **Delete Group**.

**Users**
List of users for which the Application Server is the default Application Server.

**Add User (button)**
Click to add a new user to the list of Application Server users. Clicking **Add User** opens a dialog used to select the user name.

**Delete User (button)**
Click to delete an existing Application Server user. To delete a default user, select the user name at **Users** then click **Delete User**.

**OK (button)**
Click to apply the new property settings to the current Application Server and close **Application Server Properties**.
Cancel (button)
Click to close Application Server Properties without applying any property changes to the current Application Server.

Apply (button)
Click to apply the new property settings to the current Application Server and remain in Application Server Properties.
Developer’s Guide

Introduction

A feature of the Application Server module’s design is its use of .NET assemblies. These assemblies enable custom applications developers to write code that automatically accesses Application Server module functions from within their custom applications. Known as Application Server developer objects, the Application Server module’s capabilities reside in the Microsoft Shared folder for Microsoft Dynamics SL and consist of the following components:

- **Asdevobj.dll** — A .dll providing access to the core Application Server module functions: submit Application Server requests, view request information in the Application Server queue, view Application Server processing log information, view Application Server administrative (properties and options) information.

- **Schedule.dll** — A .dll providing access to the Application Server module’s request scheduling information (when an Application Server request that requires scheduling is submitted to the request queue, Schedule.dll is used to define the scheduling information).

![Diagram of Application Server Module Developer Objects](image)

*Figure 58: Application Server Module Developer Objects*

The Application Server module developer objects are installed as part of the regular Microsoft Dynamics SL system installation. They can be used in any .NET programming language that supports the use of .NET assemblies. However, regardless of programming language, any application that uses the Application Server module developer objects must be run on a machine where Microsoft Dynamics SL is already installed and operating.
About This Developer’s Guide

This Application Server Developer's Guide provides applications developers with the concepts, explanations, and reference material needed to program custom software applications so that they can access Application Server module functions directly (from within the custom applications, themselves). This guide is divided into the following sections:

Setting Up the Developer Objects

Explains how to set up the Application Server developer objects as available Visual Basic® project references in both a Visual Basic and Visual Basic for Applications (VBA) programming environment.

System Database Connectivity and Object Initialization

Explains how to establish Microsoft Dynamics SL system database connection information and initialize the Application Server developer objects in order to integrate Application Server module capabilities into custom software applications.

Accessing Application Server Functions

Provides working code examples that show how four common Application Server module functions can be accessed from within a custom software application.

Developer’s Reference

Lists and describes the object classes, properties, methods, and events associated with the Application Server module’s Asdevobj.dll and Schedule.dll developer objects.
Setting Up the Developer Objects

Before Application Server module developer objects can be used in custom software application code, you must first set up the Application Server development environment. This involves identifying the Application Server module developer objects as available Visual Basic project references. This development environment must be established for custom applications developed using either Visual Basic or Visual Basic for Applications.

Setting Up the Objects for Visual Basic

To set up the Application Server development environment for a Visual Basic application:

1. Start Microsoft Visual Studio 2005 and begin a new Visual Basic Project, or open an existing Project.
2. Choose **Project | Add Reference** from the menu bar. The *Add Reference* window appears.

   ![Figure 59: Add Reference window](image)

3. Click the **Browse** tab, and then navigate to the Microsoft Shared\DynamicsSL folder in Program Files\Common Files (for example, C:\Program Files\Common Files\Microsoft Shared\DynamicsSL).

4. From the list of files, select ASDevObj.dll. If you want to provide scheduling information when submitting an Application Server request, also select Schedule.dll. Click **OK** to add the references to your project.

You can now begin using the Application Server developer objects in your Visual Basic application code.
Setting Up the Objects for VBA

To set up the Application Server development environment for Visual Basic for Applications (VBA):

1. Open the Visual Basic Editor from within any of the software application that supports VBA (Microsoft Word, Excel, Outlook, etc.).
   If you are using Microsoft Office 2007, click Visual Basic on the Developer tab. Otherwise, go to Tools | Macro and then select Visual Basic Editor.

2. Choose Tools | References. The References dialog box appears.

3. Under Available References, click the Microsoft Dynamics SL Application Server Objects check box.
   If the Microsoft Dynamics SL Application Server Objects check box does not appear, click Browse, go to C:\Program Files\Common Files\Microsoft Shared\DynamicsSL, select ASDevObj.tlb, and click OK. In the Available References list, find the check box and select it.

4. If you want to provide scheduling information when submitting an Application Server request, click the Microsoft Dynamics SL Schedule check box.
   If the Microsoft Dynamics SL Schedule check box does not appear, click Browse, go to C:\Program Files\Common Files\Microsoft Shared\DynamicsSL, select Schedule.tlb, and click OK. In the Available References list, find the check box and select it.

5. Click OK to close References.

You can now begin using the Application Server developer objects in your VBA application code.
System Database Connectivity and Object Initialization

Setting up the development environment enables you to use Application Server developer objects in your custom software application code. However, in order to integrate Application Server module capabilities into your custom software application, the application code must include instructions that perform two important tasks:

- Establish system database connection information
- Initialize the Application Server developer objects

“Establishing Database Connection Information” and “Initializing the Developer Objects” discuss these basic coding procedures.

Establishing Database Connection Information

Establishing the connection for the system database involves creating an instance of the S4DBConnect class and performing a call to the S4DBOpen method. Both tasks are illustrated in the following code snippet:

```vbnet
'System Database Connectivity and Object Initialization

Setting up the development environment enables you to use Application Server developer objects in your custom software application code. However, in order to integrate Application Server module capabilities into your custom software application, the application code must include instructions that perform two important tasks:

- Establish system database connection information
- Initialize the Application Server developer objects

“Establishing Database Connection Information” and “Initializing the Developer Objects” discuss these basic coding procedures.

Establishing Database Connection Information

Establishing the connection for the system database involves creating an instance of the S4DBConnect class and performing a call to the S4DBOpen method. Both tasks are illustrated in the following code snippet:

```vbnet
'Create an instance of the S4DBConnect class
Dim SysDatabase As New ASDevObj.S4DbConnect

'Obtain connection information for the System Database
SysDatabase.S4Server = "MSSQLSERV01"
SysDatabase.S4Database = "SOLSYSTEM"
SysDatabase.S4Company = "0060"
SysDatabase.S4User = "APCLERK"
SysDatabase.S4Password = ""
SysDatabase.S4DBOpen
```

You can create the SysDatabase variable either globally or within a single project procedure. If you plan to use the Application Server developer objects in more than one procedure, it is more efficient to create a global instance of the variable and then call the S4DBOpen function in the Form Load event of your project’s main form. This establishes system database connection information for the entire custom application project. You can then use the Application Server developer objects as required anywhere (that is, in any number of procedures) throughout the project.

If you are only using Application Server developer objects in a single project procedure, you can create the SysDatabase variable and perform the S4DBOpen call locally within this procedure. This reduces the number of global variables in the project. However, it does limit the use of the Application Server developer objects to the specified project procedure.

You must specify all the credentials needed to verify that a user has access to Microsoft Dynamics SL. There are new properties that must be set before the S4DBOpen method can be called. The properties and the information that is required are:

- S4Server — The name of the computer that is running Microsoft® SQL Server®
- S4Database — The name of a Microsoft Dynamics SL system database
- S4Company — The company you would like to access
- S4User — The Microsoft Dynamics SL user ID
- S4Password — The password of the Microsoft Dynamics SL user
Initializing the Developer Objects

Within a custom application project procedure, you can use either of the following methods to initialize an Application Server development object for use:

- **Method 1** — Define the variable at the beginning of the procedure and instantiate it (create an object from a class using New) as needed throughout the procedure, for example:

  ```vba
  'Define the variable to be used for submitting requests to Application Server.
  Dim ASRequest As ASDevObj.ASRequest
  'Instantiate the variable so we can use it
  Set ASRequest = New ASDevObj.ASRequest
  ```

- **Method 2** — Define and instantiate the variable all in one step, for example:

  ```vba
  'Define the variable to be used for submitting requests to Application Server.
  Dim ASRequest As New ASDevObj.ASRequest
  ```

Other methods can be used to initialize Application Server developer objects (for example, Visual Basic's Create-Object function). However, either of the above two methods is preferred.

**Note:** If you are attempting to use some aspect of VBScript (for example, an Active Server Page [ASP] or Browser script) to access Application Server developer objects, you must use the CreateObject function since VBScript does not support the New operator.
Accessing Application Server Functions

Access to Application Server module functions from custom software applications is provided by means of Application server development object classes and their associated properties, methods, and events, coded directly into the custom application code. Following is a typical section of code for accessing Application Server module functions.

Private Sub Command1_Click()
    'Create an object for managing connection to the Microsoft Dynamics SL System Database
    Dim asConn As New AsDevObj.S4DBConnect
    'Create an object to use to submit a request to Application Server
    Dim asReq As New ASDevObj.ASRequest
    'We will use this variable to store the request ID from Application Server once the request is submitted.
    Dim reqID As Integer
    'Obtain connectivity information for System database
    asConn.S4Server = "SQLSERVER01"
    asConn.S4Database = "SLSYSTEM"
    asConn.S4Company = "0060"
    asConn.S4User = "APCLEERK"
    asConn.S4Password = "Password"
    asConn.S4Dbopen
    'Specify which companies this request will be run against. In this case we will just run against the company the user is currently logged into
    asReq.AddCompany ("0060")
    'We are submitting a request to run a report so we will assume that emailAddr was previously set up with an email address where this report should be emailed and that emailText contains a string of text to be put in body of text of the email. The report will be saved in PDF format
    asReq.AddEmailDist("SYSADMIN", "TestUser @somewhere.com", "Report", ASDevObj.ASRequest.ASRDistTypeEnum.ASRDistType_AdobePDF)
    'Specify that this is a report we are submitting to Application Server
    asReq.ReqType = "R"
    'Specify a description for the request
    asReq.RequestDescription = "Submit Request using ASDEVOBJ"
    'Specify the screen number that Application Server should run for this request
    asReq.RequestName = "01660"
    'Submit the request to Application Server
    reqID = asReq.Submit
    If reqID < 0 Then
        'Error occurred, do something about it
        . . .
    Else
        'Submit succeeded.
    End If
The rest of this section provides code examples for accessing four of the Application Server module's most common functions:

- Submitting an Application Server request
- Viewing requests currently listed in the Application Server queue
- Displaying the current information in the Application Server processing log
- Viewing Application Server administrative information
- Upload a document using DocShare settings
- Upload a document to SharePoint Document Library

For a listing and explanation of the Application Server development object classes, properties, methods, and events, see “Developer’s Reference” on page 95.
Submitting an Application Server Request

The following example illustrates the coding required to submit a processing request to the Application Server queue.

Imports ASDevObj.ASRequest

Private Sub SubmitRequest()
    Dim SysDatabaseCS As New ASDevObj.S4DBConnect
    Dim AppSrvRequest As New ASDevObj.ASRequest

    SysDatabaseCS.S4Server = "SQLSERVER01"
    SysDatabaseCS.S4Database = "SLSYSTEM"
    SysDatabaseCS.S4Company = "0060"
    SysDatabaseCS.S4User = "APCLEERK"
    SysDatabaseCS.S4Password = "Password"

    'Create connection to Microsoft Dynamics SL System Database
    SysDatabaseCS.S4DBOpen

    'Everything between the With and End With will access the AppSrvRequest object
    With AppSrvRequest
        'Set up this request to run against company that the user is currently logged into.
        .AddCompany "0060"

        'Email the results of the report in PDF format to jtester@somewhere.com and ktester@somewhere.com
        .AddEmailDist("jtestrt", "jtester@somewhere.com", "Here is your report", ASDevObj.ASRequest.ASRDistTypeEnum.ASRDistType_ADOBEPDF )

        'Set the req routing flag so App Server knows to email the results
        .ReqRouting = 1

        'This is a report request so tell App Server that
        .ReqType = "R"

        'Supply a description for the request
        .RequestDescription = "Joe’s Subaccount Report"

        'Run the subaccounts (01660) report
        .RequestName = "01660"

        'Submit the request to App Server
        .Submit
    End With

    Set AppSrvRequest = Nothing
    Set SysDatabaseCS = Nothing
End Sub
Submitting a Request with Scheduling Information

Sub Submit_Scheduled()
Dim AppSrvRequest As New ASDevObj.ASRequest
Dim retval as integer
...

Your Submission code goes here. After the request is submitted, you set the schedule options via this type of code.

    retval = AppSrvRequest.Submit
    If retval < 0 Then
        'Figure out what the error is...
        Call MsgBox(AppSrvRequest.validateError(-retval), MsgBoxStyle.OkOnly, "TestApp")
    Else

        Dim timestr As String
        Dim hours As Integer
        Dim schedDB As New SolScheduler.S4DBConnect
        Dim mSchedule As SolScheduler.SchedEntry
        Dim cnt As Integer
        Dim Evts(8) As Integer
        Dim idLen As Integer

        schedDB.S4User = "UserId"
        schedDB.S4Company = "CpnyID"
        schedDB.S4Database = "DBNameSystem"
        schedDB.S4Password = "TEST"
        schedDB.S4Server = "DBServer"

        schedDB.S4DBOpen
        ' Gather the value from a Combo Box on the apps form
        Select Case cboFrequency.SelectedItem
            Case 0
                'No scheduling information specified.
                'We do not need to do anything!

            Case 1
                'One time-only scheduling
                timestr = GetTime(StartDailyTime)

                Set mSchedule = New SolScheduler.SchedEntry
                mSchedule.ID = "AS"
                mSchedule.Item = CStr(AppSrvRequest.ID)
                mSchedule.ItemType = sch_AppNotify
                mSchedule.User = bpes.UserId
                mSchedule.SetOneTimeSchedule Mid$(bAppSrvRequest.ScheduleOptions, 1, 2) & "/" & Mid$(bAppSrvRequest.ScheduleOptions, 3, 2) & "/" & Mid$(bAppSrvRequest.ScheduleOptions, 5, 4), timestr
                mSchedule.Save

            Case 2
                'Daily scheduling

        Case 3
            'Weekly scheduling

        Case 4
            'Monthly scheduling

        Case 5
            'Yearly scheduling

        End Select
    End If
End Sub
timestr = GetTime(StartDailyTime)

Set mSchedule = New SolScheduler.SchedEntry
mSchedule.ID = "AS"
mSchedule.Item = CStr(AppSrvRequest.ID)
mSchedule.ItemType = sch_AppNotify
mSchedule.User = bpes.UserId
If optDaily(0).value = True Then
  mSchedule.SetDailySchedule timestr, sch_dEveryday, 1
ElseIf optDaily(1).value = True Then
  mSchedule.SetDailySchedule timestr, sch_dWeekday, 1
Else
  mSchedule.SetDailySchedule timestr, sch_dEveryday, CInt(txtDaily.text)
End If
mSchedule.Save

Case 3
'Weekly scheduling

timestr = GetTime(StartDailyTime)

Set mSchedule = New SolScheduler.SchedEntry
mSchedule.ID = "AS"
mSchedule.Item = CStr(AppSrvRequest.ID)
mSchedule.ItemType = sch_AppNotify
mSchedule.User = bpes.UserId

Dim wkStr As String

' Place a Y in the String for each day the request should
' be executed on
For cnt = 0 To 6
  If chkWeekly(cnt).value = 1 Then
    wkStr = wkStr & "Y"
  Else
    wkStr = wkStr & "N"
  End If
Next cnt
'mSchedule.SetWeeklySchedule timestr, CInt(txtWeekly.text), wkStr
mSchedule.Save

Case 4
'Monthly scheduling

timestr = GetTime(StartDailyTime)

Set mSchedule = New SolScheduler.SchedEntry
mSchedule.ID = "AS"
mSchedule.Item = CStr(AppSrvRequest.ID)
mSchedule.ItemType = sch_AppNotify
mSchedule.User = bpes.UserId

Dim mhStr As String

'Place a Y in the string for each month that the
' request is scheduled for
For cnt = 0 To 11
    If chkMonth(cnt).value = 1 Then
        mhStr = mhStr & "Y"
    Else
        mhStr = mhStr & "N"
    End If
Next cnt

If optMonthly(0).value = True Then
    mSchedule.SetMonthlySchedule timestr, CInt(txtMonthDay.text), "", mhStr
Else
    mSchedule.SetMonthlySchedule timestr, cboMonthlyInt.text, cboMontlyDay.text, mhStr
End If
mSchedule.Save

Case 5
'Event scheduling
    timestr = GetTime(StartDailyTime)
Set mSchedule = New SolScheduler.SchedEntry
mSchedule.ID = "AS"
mSchedule.Item = CStr(AppSrvRequest.ID)
mSchedule.ItemType = sch_AppNotify
mSchedule.User = bpes.UserId
For cnt = 1 To lstSelectSched.ListCount
    idLen = InStr(1, lstSelectSched.List(cnt - 1), " ")
    Evts(cnt - 1) = CInt(Mid$(lstSelectSched.List(cnt - 1), 1, idLen))
Next cnt
Evts(cnt) = 0
mSchedule.SetEventSchedule timestr, Evts
mSchedule.Save
End Select
AppSrvRequest.SetScheduleinfo (mSchedule.Schedule_Words)
End If
End Sub

' This function gathers information from screen controls
'The screen image would be one similar to 96.020.00 screen images seen in
'this manual.
Private Function GetTime(aTime As Control) As String
    Dim hours As Integer = 0
    Dim ScheduledTime As String = ""
    ' See if the user decided to schedule the request for a specific time
    ' or for a range of times
    If (SingleTime. Checked) Then
        ' Format is HHMM in military format
        ScheduledTime = String.Format("{0:hhnn}", aTime.Text)
    Else
        '
' Format is HHMM in military format
' The format for a range of times is:
' Start time, End time, Hour interval, Minute interval
' HHMM, HHMM, HH, MM
' Format is HHMM in military format

ScheduledTime = ScheduledTime + String.Format("{0:hhnn}", rSchedtime0.Text)
ScheduledTime = ScheduledTime + ","
ScheduledTime = ScheduledTime + String.Format("{0:hhnn}", rSchedtime1.Text)
ScheduledTime = ScheduledTime + ","

If Val(cHour.Text) < 10 Then
    ScheduledTime = ScheduledTime + "0"
End If
ScheduledTime = ScheduledTime + CStr(cHour.text)
ScheduledTime = ScheduledTime + ","
If Val(cMin.Text) < 10 Then
    ScheduledTime = ScheduledTime + "0"
End If
ScheduledTime = ScheduledTime + CStr(cMin.text)
End If

GetTime = ScheduledTime

End Function
Viewing Requests in the Queue

The following example illustrates the coding required to view the processing requests currently contained in the Application Server queue.

```vba
Imports ASDevObj.ReqQueue
Public Sub DisplayQueue ()
    Dim SysDatabaseCS as New ASDevObj.S4DBConnect
    Dim asQ As New ASDevObj.ReqQueue
    Dim aReq As New ASDEVObj.ASRequest
    SysDatabaseCS.S4Server = “SQLSERVER01”
    SysDatabaseCS.S4Database = “SLSYSTEM”
    SysDatabaseCS.S4Company = “0060”
    SysDatabaseCS.S4User = “APCLEERK”
    SysDatabaseCS.S4Password = “Password”
    ‘Create connection to Microsoft Dynamics SL System Database
    SysDatabaseCS.S4DBOpen
    ‘Open the App Server queue to retrieve all records
    Call asQ.OpenQueue (ASRFilt.ASRFilt_All,””)
    ‘Get the first (current) request from the queue
    Call asQ.GetCurrent (aReq)
    While True
        ‘Display the unique id and description for the current request
        MsgBox “Request ID: “ & Cstr (aReq.id) & “, Description: “ & aReq.RequestDescription
        ‘Get the next request from the queue
        Call asQ.GetNext (aReq)
        ‘Get out if we have reached the end of the queue
        If aReq Is Nothing Then Exit While
    Wend
    Set asQ = Nothing
    Set aReq = Nothing
End Sub
```
Displaying Processing Log Information

The following example illustrates the coding required to display the information currently listed in the Application Server processing log.

Imports ASDevObj.ASLogging.ASLoggining

Public Sub DisplayLogs(asname As String)
    Dim SysDatabaseCS as New ASDevObj.S4DBConnect
    Dim asLog As New ASDEVObj.ASLogging
    Dim i as integer
    SysDatabaseCS.S4Server = "SQLSERVER01"
    SysDatabaseCS.S4Database = "SLSYSTEM"
    SysDatabaseCS.S4Company = "0060"
    SysDatabaseCS.S4User = "APCLEERK"
    SysDatabaseCS.S4Password = "Password"
    'Create connection to Microsoft Dynamics SL System Database
    SysDatabaseCS.S4DBOpen
    'Load log info for specified App Server named APPSERVER_TEST1 into collection
    asLog.LoadLogData ( AS_Log_Filter_Server, asname )
    'Loop through each entry in the log
    for i = 1 To asLog.Count
        'Put up a message box showing each entry
        MsgBox asLog.Item(i) .GetData
    Next i
    Set asLog = Nothing
    Set SysDatabaseCS = Nothing
End Sub
Viewing Administrative Information

The following example illustrates the coding required to view current Application Server administrative (options and properties) information.

Imports ASDevObj.AppServerObj.ASStatus

Public Sub DisplayAppServers ()
    Dim SysDatabaseCS as New ASDevObj.S4DBConnect
    Dim aServers As ASDEVObj.AppServers
    Dim i as integer
    Dim asStatus As String
    SysDatabaseCS.S4Server = "SQLSERVER01"
    SysDatabaseCS.S4Database = "SLSYSTEM"
    SysDatabaseCS.S4Company = "0060"
    SysDatabaseCS.S4User = "APCLEERK"
    SysDatabaseCS.S4Password = "Password"

    'Create connection to Microsoft Dynamics SL System Database
    SysDatabaseCS.S4DBOpen
    aServers = New ASDEVObj.AppServers
    'Get a collection of all define Application Servers
    aServers.Refresh
    'Iterate through the collection of Application Servers
    for i = 1 To aServers.Count
        'Set asStatus to Running, Paused, or Shut Down
        If aServers.Item(i).ServerStatus = ASStatus_Started Then
            asStatus = "Running"
        ElseIf aServers.Item(i).ServerStatus = ASStatus_Paused Then
            asStatus = "Paused"
        Else
            asStatus = "Shut Down"
        End If
        'Put up a message box with the name of the server and whether it is running or not
        MsgBox (aServers.Item(i).ServerName & " is " & asStatus )
    Next i
    Set aServers = Nothing
    Set SysDatabaseCS = Nothing
End Sub
Upload a document using DocShare settings

The following example illustrates the coding required to upload a document to a Document Library using settings configured by the DocShare feature. Application Server will access the Microsoft Dynamics SL application database and retrieve the URL and Document Library settings defined for the entity. If the entity type is set to zero (0), the Application Server knows it is a Customer ID that is being passed and knows which records to gather to find the URL for uploading; including the customer and the name of the Document Library. For more information extending DocShare capabilities see “Appendix D: Extending Doc Share’s Capabilities” in the Software Development Kit user guide.

Dim asS4DB As New ASDevObj.S4DBConnect
Dim asReq As New ASDevObj.ASRequest
Dim myid As Integer

' Pull database information from the application’s Form
asS4DB.S4Company = S4Company.Text
asS4DB.S4User = S4User.Text
asS4DB.S4Server = s4server.Text
asS4DB.S4Password = s4Password.Text
asS4DB.S4Database = S4Database.Text

asS4DB.S4DBOpen() 'Access to system database

asReq.AppServerName = "ANY" 'Let any app server send the email
asReq.AddCompany(asS4DB.S4Company)

asReq.ReqType = "R"
asReq.RequestName = "08600"
asReq.RequestDescription = "Test upload"

asReq.PrintDestination = "C:\test" + CStr(Now.Year) + "-" + CStr(Now.Month) + "-" + CStr(Now.Day) + "-" + CStr(Now.Hour) + "-" + CStr(Now.Minute) + "-" + CStr(Now.Second) + ".doc"

asReq.PrintToFile = 1

Dim EntityID As String = "C300"
Dim Entitytype As Short = 0 ' Customer
Dim DocType As Short = 1 ' AR Statement

' The following statement will cause AS to create a Word Document and upload it to the location specified for Customer C300’s Docshare settings for the AR Statement Document Type
asReq.ASRDistRecords.Add_SharePoint_Upload(0, "W", EntityID, Entitytype, DocType)

' Submit the request
myid = asReq.Submit

asS4DB = Nothing
asReq = Nothing
Upload a document to SharePoint Document Library

The following example illustrates the coding required to upload a document to a SharePoint Document Library.

Dim asS4DB As New ASDevObj.S4DBConnect
Dim asReq As New ASDevObj.ASRequest
Dim myid As Integer

' Pull database information from the application’s Form
asS4DB.S4Company = S4Company.Text
asS4DB.S4User = S4User.Text
asS4DB.S4Server = s4server.Text
asS4DB.S4Password = s4Password.Text
asS4DB.S4Database = S4Database.Text

asS4DB.S4DBOpen() 'Access to system database

asReq.AppServerName = "ANY" 'Let any app server send the email
asReq.AddCompany(asS4DB.S4Company)

asReq.ReqType = "R"
asReq.RequestName = "08600"
asReq.RequestDescription = "Test upload"

asReq.PrintDestination = "C:\test" + CStr(Now.Year) + "-" + CStr(Now.Month) + "-" + CStr(Now.Day) + "-" + CStr(Now.Hour) + "-" + CStr(Now.Minute) + "-" + CStr(Now.Second) + ".doc"

asReq.PrintToFile = 1

' The following statement will cause AS to create a Word Document and upload it to the location specified in the last parameter
asReq.ASRDistRecords.Add_SharePoint_Upload(0, "W", "http://MyServer/My Doc Lib" )

' Submit the request
myid = asReq.Submit

asS4DB = Nothing
asReq = Nothing
Developer’s Reference

The Application Server module’s developer objects consist of the following components:

- **Asdevobj.dll** — An assembly providing access to the core Application Server module functions: submit Application Server requests, view request information in the Application Server queue, view Application Server processing log information, view Application Server administrative (properties and options) information. This assembly can only be referenced from 32-bit processes.

- **Microsoft.Dynamics.SL.AS.Devobj.dll** — A replica of the ASDevobj.dll; however it is compiled as an AnyCPU assembly. This means that this assembly can be referenced by 64-bit processes. This assembly has a majority of the same classes, properties and methods as asdevobj.dll (It is missing the Ecommunication class)

- **Microsoft.Dynamics.Communications.dll** – An assembly for email specific processing. This file is compiled as AnyCpu and can be called from 32-bit or 64-bit processes.

- **Schedule.dll** — Provides access to the Application Server module’s request scheduling information (when an Application Server request that requires scheduling is submitted to the request queue, Schedule.dll is used to define the scheduling information). This assembly can only be referenced from 32-bit processes.

- **Microsoft.Dynamics.SL.AS.Schedule.dll** – An assembly for scheduling Application Server requests. This assembly is compiled as AnyCpu and can be called from 32-bit or 64-bit processes.
Asdevobj.dll and Microsoft.Dynamics.SL.AS.DevObj.dll

These assemblies handle the submission and management of the Application Server. Asdevobj.dll is a 32-bit implementation. Microsoft.Dynamics.SL.AS.DevObj is compiled as AnyCpu and can be referenced by 64-bit processes.

The Application Server module’s Asdevobj.dll development object component consists of the following object classes:

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppServerObj</td>
<td>Represents a single Application Server.</td>
</tr>
<tr>
<td>AppServers</td>
<td>A collection of all defined Application Servers.</td>
</tr>
<tr>
<td>AppSrvDetailObj</td>
<td>A collection of users and/or groups that can submit requests to a specific Application Server.</td>
</tr>
<tr>
<td>ASLogEntry</td>
<td>A single entry in the Application Server log.</td>
</tr>
<tr>
<td>ASLogging</td>
<td>Represents log data for a specific Application Server or all Application Servers.</td>
</tr>
<tr>
<td>ASRDBase</td>
<td>A single company that a request will be run against (part of ASRequest).</td>
</tr>
<tr>
<td>ASRDbCol</td>
<td>A collection of companies that a request will be run against (part of ASRequest).</td>
</tr>
<tr>
<td>AsrDistList</td>
<td>A single user that a report should be distributed to (part of ASRequest).</td>
</tr>
<tr>
<td>AsrDistLists</td>
<td>A collection of all users that a report will be emailed to (part of ASRequest).</td>
</tr>
<tr>
<td>ASRequest</td>
<td>Represents an Application Server request.</td>
</tr>
<tr>
<td>ASRTIRecs</td>
<td>A collection of screens included in a Transaction Import request.</td>
</tr>
<tr>
<td>Ecommunication</td>
<td>General purpose email class. (Not part of Microsoft.Dynamics.SL.AS.DevObj. See Microsoft.Dynamics.SL.Communications)</td>
</tr>
<tr>
<td>ReqQueue</td>
<td>Represents the Application Server queue of requests.</td>
</tr>
<tr>
<td>S4DBConnect</td>
<td>General purpose database connectivity class. Used to connect to Microsoft Dynamics SL system database.</td>
</tr>
</tbody>
</table>
**Object Class Hierarchy**

Within Asdevobj.dll, the classes observe the following object hierarchy:

- The object classes ASRequest and AppServers are at the highest level.
- ASRDbCol, ASRDistLists, ASRTIRecs, and AppServerObj are in the middle of the object class hierarchy.
- ASRDBase, ASRDistList, and AppSrvDetailObj are at the lowest hierarchical level.

![Diagram of object class hierarchy]

*Figure 61: ASRequest object class hierarchy*

*Figure 62: AppServers object class hierarchy*

The rest of this section details the Asdevobj.dll object classes and their associated properties, methods, and events.
AppServerObj

Object class representing a single Application Server. Servers can be created, and their configurations changed, using this object.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteReports</td>
<td>Determines whether the Application Server deletes reports after they have been emailed or keeps them. A value of 1 causes the server to delete reports; 0 causes the server to keep them.</td>
</tr>
<tr>
<td>EmailProfile</td>
<td>The email profile ID that a server uses to access the email system. Application Server uses the Microsoft Visual Studio Tools for Office (VSTO) for email purposes, so this value is the Exchange or Outlook Profile ID.</td>
</tr>
<tr>
<td>IncompleteReq</td>
<td>What action the Application Server should take when it finds a request in the queue that started, but did not finish, processing.</td>
</tr>
<tr>
<td>LogDays</td>
<td>The number of days worth of log data the Application Server should store before it starts overwriting the log.</td>
</tr>
<tr>
<td>LogMaxSize</td>
<td>The maximum size (in records) that the log can grow to before the Application Server begins overwriting records.</td>
</tr>
<tr>
<td>MaximumApps</td>
<td>The Maximum number of processes AS will carry out at once. Currently fixed at 1.</td>
</tr>
<tr>
<td>PollQInterval</td>
<td>How often the Application Server polls the queue for outstanding requests; specified in seconds.</td>
</tr>
</tbody>
</table>
| ProcessEmail | Determines what type of requests can be submitted via email.  
0 – Nothing  
1 – Only Reports  
2 – All requests |
| ProcessSysAdmin | Determines if emails from the SYSADMIN user will be processed.  
0 – No  
1 – Process the email based on setting of ProcessEmail. |
| RequestLimit | The maximum number of completed requests that can be in the queue before the Application Server starts deleting requests automatically. |
| ServerName | The name of the Application Server. |
| ServerStatus | Current status of the server. |
| TempDirectory | Path to the directory used by the Application Server to store reports to be routed, to the Transaction Import files, and to other temporary data. |

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add (asdet as appsrvdetaiobj; akey as string)</td>
<td>Used to add a user or group to the list of users and groups that can submit requests to the Application Server.</td>
</tr>
<tr>
<td>Count()</td>
<td>Returns the number of users or groups that can submit requests to the Application Server.</td>
</tr>
<tr>
<td>DeleteServer()</td>
<td>Deletes the Application Server.</td>
</tr>
<tr>
<td>IsDefault(user as string)</td>
<td>Returns true if the Application Server is the default Application Server for the specified user or group.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsUserAssigned(user as string)</td>
<td>Returns true if the user or group specified is allowed to submit requests to the Application Server.</td>
</tr>
<tr>
<td>Item(ix)</td>
<td>Returns an AppSrvDetailObj object.</td>
</tr>
<tr>
<td>Remove(ix)</td>
<td>Removes an AppSrvDetailObj object from the collection.</td>
</tr>
<tr>
<td>RemoveAll</td>
<td>Removes all AppSrvDetailObj objects from the collection.</td>
</tr>
<tr>
<td>Update()</td>
<td>Update the server's configuration information in the database.</td>
</tr>
</tbody>
</table>

### AppServers

Object class representing all Application Servers that are defined for Microsoft Dynamics SL.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(apps as appserverobj)</td>
<td>Used to add an Application Server to the collection.</td>
</tr>
<tr>
<td>Count()</td>
<td>Returns the number of Application Servers in the collection.</td>
</tr>
<tr>
<td>IsUserAssigned(user as string, asname as string)</td>
<td>Returns True if the specified user can submit requests to the specified Application Server.</td>
</tr>
<tr>
<td>Item(key)</td>
<td>Returns an AppServerObj object representing the specified server.</td>
</tr>
<tr>
<td>Refresh()</td>
<td>Refreshes the collection with data from the database.</td>
</tr>
<tr>
<td>Remove(key)</td>
<td>Remove the current Application Server from the collection.</td>
</tr>
<tr>
<td>UserAssigns(user a string)</td>
<td>Creates a collection of Application Servers that the specified user can submit requests to.</td>
</tr>
</tbody>
</table>

### AppSrvDetailObj

Object class representing a specific user or group that can submit requests to an Application Server.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASDefault</td>
<td>Determines whether the current Application Server is the default Application Server for the user or group that this object represents.</td>
</tr>
<tr>
<td>Company</td>
<td>Not used.</td>
</tr>
<tr>
<td>ObjectID</td>
<td>User ID or Group ID that this object represents.</td>
</tr>
<tr>
<td>ObjectType</td>
<td>“U” for User or “G” for Group.</td>
</tr>
<tr>
<td>ServerName</td>
<td>The name of the Application Server that this assignment is for.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
**ASLogEntry**

Object class representing a single entry in the Application Server log.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetData()</td>
<td>Returns the entry data in a string.</td>
</tr>
<tr>
<td>GetDate()</td>
<td>Returns a string containing the date the entry was created.</td>
</tr>
<tr>
<td>GetID()</td>
<td>Returns the Request ID that this entry is associated with.</td>
</tr>
<tr>
<td>GetLogID()</td>
<td>Returns the unique ID for this entry within the logging system.</td>
</tr>
<tr>
<td>GetServer()</td>
<td>Returns the name of the Application Server that created this log entry.</td>
</tr>
<tr>
<td>SetEntry(logid as long, id as long, server as string, data as string)</td>
<td>Sets the data associated with this entry. Used for saving log entries into the log.</td>
</tr>
</tbody>
</table>

**ASLogging**

Object class representing the logging system; used for creating and retrieving entries from the Application Server log.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(asname as string, id as integer, LogText as string)</td>
<td>Creates a new log entry in the Application Server log.</td>
</tr>
<tr>
<td>Count()</td>
<td>Returns the number of entries in the current log.</td>
</tr>
<tr>
<td>Delete(ix)</td>
<td>Deletes the associated entry (as specified by ix) from the collection of log entries.</td>
</tr>
<tr>
<td>DeleteLogData(filtertype as integer, filtervalue)</td>
<td>Deletes log entries based on the criteria specified.</td>
</tr>
<tr>
<td>GetServers()</td>
<td>Returns a Visual Basic collection containing the names of all Application Servers for which there are entries in the log.</td>
</tr>
<tr>
<td>Item(ix)</td>
<td>Retrieves the associated entry (as specified by ix) from the collection of log entries. Returns an ASLogEntry object.</td>
</tr>
<tr>
<td>LoadLogData(filtertype as integer, filtervalue)</td>
<td>Load log data from the database into the ASLogging object based on the criteria specified.</td>
</tr>
<tr>
<td>ReadLogData(fname as string)</td>
<td>Reads the archived log file into the ASLogging object.</td>
</tr>
<tr>
<td>SaveLogData(filtertype as integer, filtervalue, fname as string)</td>
<td>Saves the log data, based on the criteria specified, into an archive file.</td>
</tr>
</tbody>
</table>
## ASRDBase

Object class representing a single company that an Application Server request can run against.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompanyID</td>
<td>The company ID that this object represents.</td>
</tr>
<tr>
<td>ExecLocation</td>
<td>Must be 1.</td>
</tr>
<tr>
<td>Id</td>
<td>The request ID from ASRequest that this object is associated with.</td>
</tr>
<tr>
<td>Instance</td>
<td>Represents a Windows Instance. Must be 0.</td>
</tr>
<tr>
<td>OutputInfo</td>
<td>Contains the name of any event logs created when a request is run against the company represented by this object.</td>
</tr>
<tr>
<td>Status</td>
<td>Indicates whether the request has been run against this company or not.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update()</td>
<td>Updates the entry in the database with the information in this object.</td>
</tr>
</tbody>
</table>

## ASRDbCol

Object class representing a collection of the companies a particular request can run against.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(id as long, companyid as string, status as integer, execloc as integer, outputinfo as string)</td>
<td>Adds a new company to the collection.</td>
</tr>
<tr>
<td>Count()</td>
<td>Returns the number of entries in the collection. This is the number of companies that the associated request can run against.</td>
</tr>
<tr>
<td>Delete(index)</td>
<td>Deletes the associated (as specified by index) company from the collection.</td>
</tr>
<tr>
<td>Item(index)</td>
<td>Returns the associated (as specified by index) company from the collection. Returns an ASRDBase object.</td>
</tr>
<tr>
<td>Submit()</td>
<td>Saves the collection of companies to the database queue.</td>
</tr>
</tbody>
</table>

## ASRDistList

Object class representing an email recipient to receive reports distributed by an Application Server when the associated request is processed.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DistBody</td>
<td>Message that the Application Server should put in the body text of the email to be sent.</td>
</tr>
<tr>
<td>DistEmailAddr</td>
<td>Email address this entry represents.</td>
</tr>
<tr>
<td>DistRptType</td>
<td>Format the report should be saved in (Word, WordPerfect, Excel, Lotus, Text, CSV, HTML, etc.).</td>
</tr>
<tr>
<td>DistStatus</td>
<td>Status of this entry.</td>
</tr>
<tr>
<td>DistUser</td>
<td>ID of the Microsoft Dynamics SL user that this entry represents. May be blank.</td>
</tr>
</tbody>
</table>
FaxNbr | The fax number to be used to fax the associated document.
FaxReceiverName | The name of the person or company that will be receiving the Fax.
RequestID | Unique request ID indicating which Application Server request this entry is associated with.
SLObjectID | The Entity ID of a DocShare Entity. This typically a Customer ID, Vendor ID or Project ID.
SLObjectType | This value defines the entity type to which the SLObject ID belongs. For example, is it a Vendor ID or a Customer ID.
SLeodocumentType | This defines what type of DocShare document is going to be distributed.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Update()</td>
<td>Updates the entry in the database with the information in this object.</td>
</tr>
</tbody>
</table>

**ASRDistLists**

Object class representing a collection of all recipients to which an email report should be sent.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add(id as long, bodytext as string, userid as string, emailaddr as string, rpttype as string)</td>
<td>Adds a new email entry to the collection.</td>
</tr>
<tr>
<td>Add_SharePoint_Upload (ID as Integer, FileType as String, DocLib as string)</td>
<td>Adds a SharePoint document library as a destination. This will upload the file (specified in the ASRequest Object) of the specified type (FileType) to the specified Document Library (DocLib).</td>
</tr>
<tr>
<td>Add_SharePoint_Upload(ID as Integer, FileType as String, EntityID as String, Entitytype as short, DocType as short)</td>
<td>Adds a SharePoint document library as a destination. This will upload the file (specified in the ASRequest Object) of the specified type (FileType) to the Document Library that is configured in DocShare Setup for the specified Entity and DocType. (see example for more details)</td>
</tr>
<tr>
<td>Count()</td>
<td>Returns the number of email address entries in the collection.</td>
</tr>
<tr>
<td>Delete(ix)</td>
<td>Deletes the associated (based on ix) email entry from the collection.</td>
</tr>
<tr>
<td>Item(index)</td>
<td>Returns the associated (based on index) email entry from the collection. Returns an ASRDistList object.</td>
</tr>
<tr>
<td>Submit()</td>
<td>Saves the collection of entries to the database queue.</td>
</tr>
</tbody>
</table>

**ASRequest**

Object class representing an Application Server request.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AppServerName</td>
<td>The name of the server to run this request when submitting or viewing a request waiting in the queue. The name of the server that did run the request when viewing a completed request.</td>
</tr>
<tr>
<td>Properties</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ASRDbRecords</td>
<td>Provides access to the ASRDbCol object containing one entry for each company this request will be or has been run against.</td>
</tr>
<tr>
<td>ASRDistRecords</td>
<td>Provides access to the ASRDistLists object containing one entry for each email address to which a report will be sent.</td>
</tr>
<tr>
<td>ExecLocation</td>
<td>Must be 1.</td>
</tr>
<tr>
<td>ExecuteID and</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>ExecutePassword</td>
<td></td>
</tr>
<tr>
<td>GenLocation</td>
<td>Must be 1.</td>
</tr>
<tr>
<td>Id</td>
<td>The unique ID associated with this request.</td>
</tr>
<tr>
<td>MasterID</td>
<td>Associates a scheduled request with its master request.</td>
</tr>
<tr>
<td>MaxPages</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>Notification</td>
<td>0 – No notification; 8 – Notify when request is completed (see RequestorsEmail property).</td>
</tr>
<tr>
<td>OutputFormat</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>OutputList</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>Password</td>
<td>Password associated with request. Do not alter.</td>
</tr>
<tr>
<td>Printxxxxxxx</td>
<td>The various print properties used to set up printer information for report requests.</td>
</tr>
<tr>
<td>PrintDestination</td>
<td>The name of the file to create when exporting to a file</td>
</tr>
<tr>
<td></td>
<td>Or</td>
</tr>
<tr>
<td></td>
<td>The name of the printer to print the report</td>
</tr>
<tr>
<td>PrintDevice</td>
<td>The name of the printer device</td>
</tr>
<tr>
<td></td>
<td>Or</td>
</tr>
<tr>
<td></td>
<td>The URL of an existing SharePoint Document Library</td>
</tr>
<tr>
<td>PrintDriver</td>
<td>The name of the printer device</td>
</tr>
<tr>
<td></td>
<td>Or</td>
</tr>
<tr>
<td></td>
<td>“SLSP” if you want ROI to upload the exported report to a SharePoint Document Library (specified in PrintDevice)</td>
</tr>
<tr>
<td>PrintToFile</td>
<td>Set to 1 if you would like ROI to export the report to a file. Specify the file type in the TIErrors property and the filename in the PrintDestination Property.</td>
</tr>
<tr>
<td>Priority</td>
<td>Indicates the current priority of the request: 4 = Scheduled, 3 = High, 2 = Normal, 1 = Low.</td>
</tr>
<tr>
<td>RemoteID</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>ReqParms</td>
<td>Command line parameters to pass to the requested screen when it is run.</td>
</tr>
<tr>
<td>ReqRouting</td>
<td>Flag indicating whether the request included report distribution information.</td>
</tr>
<tr>
<td>ReqType</td>
<td>Type of request: R = Report; P = Process.</td>
</tr>
<tr>
<td>RequestDescription</td>
<td>A description to be associated with the request.</td>
</tr>
<tr>
<td>RequestName</td>
<td>The name of the screen to run for this request; 5 characters long (for example, Subaccounts Report is 01660).</td>
</tr>
<tr>
<td>RequestorsEmail</td>
<td>Email addresses to send notifications to when a request is complete. Separate addresses with semicolons.</td>
</tr>
<tr>
<td>RequestStatus</td>
<td>The current status of the request: 0 = Waiting in Queue; 1 = Retrieved from Queue; 2 = Processing; 3 = Complete; 4 = Sent; 5 = Security Violation; 6 = Incomplete, 16 = Hold.</td>
</tr>
<tr>
<td>RequestTime</td>
<td>The time and date when a request completed processing.alement of the printer device Or “SLSP” if you want ROI to upload the exported report to a SharePoint Document Library (specified in PrintDevice)</td>
</tr>
</tbody>
</table>
### Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ScheduleOptions</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>Template</td>
<td>Template to apply to the screen when it is run. Template represented in string format.</td>
</tr>
<tr>
<td>TemplateId</td>
<td>ID of a template to apply to the screen when it is run.</td>
</tr>
<tr>
<td>TIErrors</td>
<td>For a Transaction Import request, the maximum number of errors that the Transaction Import can encounter before it will stop processing. OR A value that specifies the file type for ROI will export to.</td>
</tr>
<tr>
<td>TILogInfo</td>
<td>Transaction Import logging options.</td>
</tr>
<tr>
<td>TIProcessing</td>
<td>Transaction Import processing options.</td>
</tr>
<tr>
<td>UserId</td>
<td>ID of the Microsoft Dynamics SL user the request will be, or was, run for.</td>
</tr>
<tr>
<td>VBButtonName</td>
<td>Name of the button on the RequestName screen that Application Server should push to make the process or report run. By default, this is cBegProcessing.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddCompany(cid as string)</td>
<td>Add a company ID to run this request against.</td>
</tr>
<tr>
<td>AddEmailDist(userid as string, emailaddr as string, emailltext as string, rptype as ASRDistTypeEnum)</td>
<td>Add a recipient to the list of users to receive email reports.</td>
</tr>
<tr>
<td>AddTIRec(id as long, dtafile as string, ctlfile as string, outfile as string, screen as string, data as string)</td>
<td>Add a new record to be processed by Transaction Import when Application Server carries out a Transaction Import request.</td>
</tr>
<tr>
<td>GetData()</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>IsValid()</td>
<td>Returns True if the request is valid; False if any errors are encountered.</td>
</tr>
<tr>
<td>NotifyMe()</td>
<td>Ask to be notified, via an event, when the request associated with this object is complete.</td>
</tr>
<tr>
<td>PropValue(cnt as long)</td>
<td></td>
</tr>
<tr>
<td>QkTimer</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>SetASid</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>SetData</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>SetStatus(status as integer)</td>
<td>Set the request status.</td>
</tr>
<tr>
<td>Submit()</td>
<td>Submit the request and sets the status to Waiting In Queue.</td>
</tr>
<tr>
<td>Submit(Status as integer )</td>
<td>Submit the request and set the status to specified value.</td>
</tr>
<tr>
<td>TIItem(ix as integer)</td>
<td>Retrieve a particular Transaction Import record, as indicated by ix.</td>
</tr>
<tr>
<td>TIRecordCount()</td>
<td>Returns the number of Transaction Import records associated with this request.</td>
</tr>
<tr>
<td>Validate()</td>
<td>Check whether all required properties have valid values.</td>
</tr>
<tr>
<td>ValidateError()</td>
<td>Currently unsupported.</td>
</tr>
</tbody>
</table>
### Events

<table>
<thead>
<tr>
<th>Description</th>
<th>Event Indication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event indicating the status of the request has changed.</td>
<td>EvCode indicates the reason for the event being fired.</td>
</tr>
</tbody>
</table>

### ASRTIRecs

Object class representing all the screens that the Transaction Import module should be run against for this request.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControlFile</td>
<td>Name of the control macro to use for the associated screen.</td>
</tr>
<tr>
<td>DataFile</td>
<td>Name of the data file to use for the associated screen.</td>
</tr>
<tr>
<td>FileData</td>
<td></td>
</tr>
<tr>
<td>OutputFile</td>
<td>Name of the output file to be generated when Transaction Import is run.</td>
</tr>
<tr>
<td>RequestID</td>
<td>The ID of the request that this Transaction Import record is associated with.</td>
</tr>
<tr>
<td>ScreenNum</td>
<td>The number of the screen that this record represents.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submit()</td>
</tr>
<tr>
<td>Save the data associated with this object to the database.</td>
</tr>
</tbody>
</table>

### ECommunication

Object class used for accessing, sending, and receiving, email. Uses Microsoft Visual Studio Tools for Office (VSTO) for email capabilities.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommAddress</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>CommPassword</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>ProfileID</td>
<td>Name of Outlook/Exchange profile ID to be used for logging into email.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseMessages()</td>
</tr>
<tr>
<td>Currently has no effect.</td>
</tr>
<tr>
<td>GetMessages()</td>
</tr>
<tr>
<td>Returns a Outlook items object containing all messages in the Inbox.</td>
</tr>
<tr>
<td>SendMsg(theReceiver() as string, theMsg as variant, numAtt as integer, Att() as string, [theSubject])</td>
</tr>
<tr>
<td>Send an email to the specified email address, including the subject specified, the body of text specified, and the attachments specified.</td>
</tr>
<tr>
<td>StartUp()</td>
</tr>
<tr>
<td>Start the email service using the Outlook/Exchange profile specified.</td>
</tr>
</tbody>
</table>

### ReqQueue

Object class representing the Application Server Request Queue.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>
## Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CopyRequest(rid as long)</td>
<td>Copies the request specified to a new request.</td>
</tr>
<tr>
<td>CountWaiting(asname as string, waitType as integer)</td>
<td>Returns the number of requests in the queue waiting to be run that are eligible to be run by the Application Server specified.</td>
</tr>
<tr>
<td>DeleteRequest(id as long)</td>
<td>Deletes the specified request from the queue.</td>
</tr>
<tr>
<td>GetByID(id as long)</td>
<td>Retrieves a request from the queue.</td>
</tr>
<tr>
<td>GetByMasterID(id as long)</td>
<td>Retrieves a request from the queue using the Master ID.</td>
</tr>
<tr>
<td>GetCommand(asname as string, requestobj as asrequest)</td>
<td>Retrieves the first request in the queue that is a command.</td>
</tr>
<tr>
<td>GetCurrent(areq as asrequest)</td>
<td>Gets the information for the current request from the database.</td>
</tr>
<tr>
<td>GetNext(asrequest)</td>
<td>Retrieves the request from queue following the current request.</td>
</tr>
<tr>
<td>GetPrevious(asreq as asrequest)</td>
<td>Retrieves the request from queue preceding the current request.</td>
</tr>
<tr>
<td>GetProcessing(asname as string, asreq as asrequest)</td>
<td>Retrieves the first request from the queue that has a status of processing.</td>
</tr>
<tr>
<td>GetRequest(acmd as command, asreq as asrequest, statusflag as integer)</td>
<td>Retrieves a request from the queue for processing by an Application Server.</td>
</tr>
<tr>
<td>GetScheduled()</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>GetWaiting(asname as string, asreq as asrequest)</td>
<td>Retrieves the first request from the queue that is waiting to be processed.</td>
</tr>
<tr>
<td>OpenQueue(rTyp e as ASRFilt, rValue)</td>
<td>Opens the queue. Selects records from the queue based on the filter information specified.</td>
</tr>
<tr>
<td>PurgeRequests(asname as string, purgedate as string)</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>SchedRequest(rid as long, mid)</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>SetASid()</td>
<td>For internal use only.</td>
</tr>
<tr>
<td>SetOwner(own as string)</td>
<td>Name of the application Server using the queue.</td>
</tr>
<tr>
<td>Update(updreq as asrequest)</td>
<td>Updates the data in the database using the data from the current object.</td>
</tr>
</tbody>
</table>

## S4DBConnect

General purpose object class used to access the Microsoft Dynamics SL system database.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsConnected</td>
<td>True if a connection has been established; False otherwise.</td>
</tr>
<tr>
<td>S4Company</td>
<td>Name of company to log into.</td>
</tr>
<tr>
<td>S4Database</td>
<td>Name of System database to log into.</td>
</tr>
<tr>
<td>S4Password</td>
<td>Microsoft Dynamics SL user’s password.</td>
</tr>
<tr>
<td>S4Server</td>
<td>Name of database server.</td>
</tr>
<tr>
<td>S4User</td>
<td>Microsoft Dynamics SL user ID.</td>
</tr>
<tr>
<td>Methods</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>S4DBOpen()</td>
<td>Opens the specified database. If no properties have been set in advance, this method opens the current default database and company.</td>
</tr>
</tbody>
</table>
Schedule.dll and Microsoft.Dynamics.SLAS.Schedule.dll

These assemblies handle the scheduling of requests. Schedule.dll is a 32-bit implementation. Microsoft.Dynamics.SLAS.Schedule is compiled as AnyCpu and can be referenced by 64-bit processes.

The Application Server module’s Schedule.dll development object component consists of the following object classes:

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SchedEntry</td>
<td>Represents one entry in the scheduler database.</td>
</tr>
<tr>
<td>SchedEntries</td>
<td>A collection of entries from the scheduler database.</td>
</tr>
<tr>
<td>SchedNotify</td>
<td>This class is used by an application to receive notifications whenever a scheduled item is ready to be processed.</td>
</tr>
<tr>
<td>S4DBConnect</td>
<td>General purpose database connectivity class. Used to connect to the system database.</td>
</tr>
</tbody>
</table>

The following are defined for use with some of the methods supported by the objects.

Object Class Hierarchy

Within Schedule.dll, the classes observe the following object hierarchy:

- The object classes SchedEntries and SchedNotify are at the highest level.
- SchedEntry is at the lowest hierarchical level.

![Figure 63: Schedule.dll object hierarchy](image)

SchDailyIntervalEnum

Object class used to indicate how often an entry scheduled on a daily basis should be processed. Possible values are:

- Sch_dEveryday: Used to indicate that an entry that is scheduled on a daily basis should be run every day.
- Sch_dWeekday: Used to indicate that an entry that is scheduled on a daily basis should be run on weekdays only.
SchEntries
Object class used to determine which entries from the scheduler database should be returned from a GetEntries call. Possible values are:
- **SchE_All**: Indicates that all entries should be retrieved during a GetEntries call.
- **SchE_UID**: Indicates that all entries matching the ID specified should be retrieved during a GetEntries call.
- **SchE_User**: Indicates that all entries for the specified user should be retrieved.

The rest of this section details the Schedule.dll and Microsoft.Dynamics.SLAS.Schedule.dll object classes and their associated properties, methods, and events.

SchedEntry
Object class representing a single entry in the Scheduler system.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Active</strong></td>
<td>Determines whether the entry is an active schedulable entry or not. TRUE means the entry is active; FALSE means the entry is not active.</td>
</tr>
<tr>
<td><strong>ID</strong></td>
<td>An ID representing what the scheduled item is a part of. For Application Server requests this value should be “AS.”</td>
</tr>
<tr>
<td><strong>Interval</strong></td>
<td>The scheduling interval that determines how often the entry should be processed.</td>
</tr>
<tr>
<td><strong>Item</strong></td>
<td>An ID representing what is being scheduled that the application using the scheduler can use for its own purposes. For Application Server requests the Item property should be set to the string representation of the unique Request ID generated when a request is created.</td>
</tr>
<tr>
<td><strong>ItemType</strong></td>
<td>Determines whether the entry is for an executable that the scheduler should run or if the scheduler should notify a separate application when an entry is ready to be processed. For Application Server requests this value must always be sch_AppNotify indicating that Application Server will be notified when the entry is ready to be processed.</td>
</tr>
<tr>
<td><strong>LastExec</strong></td>
<td>Represents the last date when the entry was processed.</td>
</tr>
<tr>
<td><strong>Sched_Hour</strong></td>
<td>The hour interval used when submitting a request to run several times in one day</td>
</tr>
<tr>
<td><strong>Sched_Min</strong></td>
<td>The minute interval used when submitting a request to run several times in one day</td>
</tr>
<tr>
<td><strong>Sched_Start</strong></td>
<td>The Start of a period of time used when submitting a request to run several times in one day</td>
</tr>
<tr>
<td><strong>Sched_End</strong></td>
<td>The End of a period of time used when submitting a request to run several times in one day</td>
</tr>
<tr>
<td><strong>ScheduleInfo</strong></td>
<td>String representation of the actual scheduling information. This value should be set using one of the methods (Set…) described below.</td>
</tr>
<tr>
<td><strong>StartDate</strong></td>
<td>Next date that this entry should be processed.</td>
</tr>
<tr>
<td><strong>StartTime</strong></td>
<td>Time when this entry should be processed on the date specified by StartDate.</td>
</tr>
<tr>
<td><strong>TimeRangeSched</strong></td>
<td>True or False value stating whether this request will be run several times in one day.</td>
</tr>
<tr>
<td><strong>UniqueID</strong></td>
<td>ID that uniquely represents this entry in the database table. Generated by the database.</td>
</tr>
<tr>
<td>Properties</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>User</td>
<td>Microsoft Dynamics SL user that this entry is scheduled on behalf of.</td>
</tr>
<tr>
<td>Value</td>
<td>Scheduling interval information. This value should be set using one of the methods (Set...) described below.</td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete()</td>
<td>Should be called any time an entry is processed. This is used by the scheduler to determine the next date and time an entry will be processed.</td>
</tr>
<tr>
<td>Delete()</td>
<td>Removes the entry from the schedulers list of items.</td>
</tr>
<tr>
<td>IsActive()</td>
<td>Determines whether an entry should be processed or not. Returns TRUE if the entry is active and can be processed; returns FALSE if not.</td>
</tr>
<tr>
<td>Save()</td>
<td>Saves this entry to the database. This should be used the first time an entry is created.</td>
</tr>
<tr>
<td>SetDailySchedule()</td>
<td>Sets the entry to be processed on a daily basis.</td>
</tr>
<tr>
<td>SetEventSchedule()</td>
<td>Sets the entry to be processed based on other scheduled entries completing.</td>
</tr>
<tr>
<td>SetMonthlySchedule()</td>
<td>Sets the entry to be scheduled on a monthly basis.</td>
</tr>
<tr>
<td>SetOneTimeSchedule()</td>
<td>Sets the entry to be scheduled one time only, at the date and time specified. The date must be in the format of MM/DD/YYYY.</td>
</tr>
<tr>
<td>SetWeeklySchedule()</td>
<td>Sets the entry to be scheduled on a weekly basis.</td>
</tr>
<tr>
<td>Update(sched as Boolean)</td>
<td>Updates the schedule entry in the database with the current information. Sched should always be set to FALSE.</td>
</tr>
</tbody>
</table>

### SchedEntries

Object class representing a collection of schedule entries.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count()</td>
<td>Returns the number of entries in the collection.</td>
</tr>
<tr>
<td>DeleteEntry(ix as Integer)</td>
<td>Deletes an entry from the database.</td>
</tr>
<tr>
<td>DoneSched(ix as integer)</td>
<td>Should be called after an entry has been processed. This is automatically called by Application Server when a request is processed.</td>
</tr>
<tr>
<td>GetDailyEntries(type as schEntries, [item as string])</td>
<td>Retrieves all of the schedule entries that will need to be processed during the current day, based on the criteria specified.</td>
</tr>
<tr>
<td>GetEntries(type as schEntries, [schItem as string])</td>
<td>Retrieves all the schedule entries that match the criteria specified.</td>
</tr>
</tbody>
</table>
Methods Description
IsCurrentDate() Returns TRUE if the entries in the collection are to be processed on the current day; returns FALSE if there are no entries in the collection or they are not scheduled to be processed on the current day.
Item(ix as Integer) Returns a schedule entry from the collection; ix is an index into the collection which determines which entry to return.
Remove(ix as Integer) Deletes an entry from the current collection. Unlike DeleteEntry, this method does not delete the entry from database, only from the current collection.
RetrieveEntries(type as schEntries item as string, sqls as string) Read entries from the database into the collection based on criteria specified.

SchedNotify
Object class used by an application when it wants to be notified when items need to be processed based on the scheduling information. Application Server uses this to receive notification any time a schedule request needs to be processed.

Properties Description
IsNotify Returns TRUE if the object is set up to return notification; returns FALSE if it is not set up to return notifications.

Methods Description
Activate(type as schNotify, item as string) Used to turn on notifications. Parameters specify which types of items the object should post notifications for.
DeActivate() Stops notifications.
SchedComplete(item as string) This should be called anytime a notification has been received and the specified item has been processed.

Events Description
Scheduled(entry as SchedEntry) This event fires when notifications are turned on and an entry matching the parameters specified in the Activate call is ready to be processed based on the scheduling information contained in the entry.

S4DBConnect
General purpose object class used to access Microsoft Dynamics SL system database.

Properties Description
IsConnected True if a connection has been established; false otherwise.
S4Company Name of company to log into.
S4Database Name of System Database to log into.
S4Password The password of a Microsoft Dynamics SL user.
S4Server Name of database server.
S4User Microsoft Dynamics SL user ID.

Methods Description
S4DBOpen() Open the specified database. If no properties have been set in advance then this method opens the current default database and company.
Microsoft.Dynamics.SL.Communications
An assembly used for accessing, sending, and receiving, email. Uses a Visual Studio for Office Outlook Addin for email capabilities. This Assembly is compiled as AnyCpu and can be referenced by a 64-bit process.

ECommunication
Object class used for accessing, sending, and receiving, email.

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommAddress</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>CommPassword</td>
<td>Currently unsupported.</td>
</tr>
<tr>
<td>ProfileID</td>
<td>Name of Outlook/Exchange profile ID to be used for logging into email.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloseMessages()</td>
<td>Closes the connection</td>
</tr>
<tr>
<td>GetMessages()</td>
<td>Returns an object containing all messages in the In box.</td>
</tr>
<tr>
<td>SendMsg( theReceiver() as string, theMsg as variant, numAtt as integer, Att() as string, [theSubject])</td>
<td>Send an email to the specified email address, including the subject specified, the body of text specified, and the attachments specified.</td>
</tr>
<tr>
<td>StartUp()</td>
<td>Start the email service using the Outlook or Exchange profile specified.</td>
</tr>
</tbody>
</table>
Glossary

Application Server
The server that processes requests sent from a client workstation.

Application Server Queue
The list of requests waiting to be processed by the Application Server.

Client
The workstation where requests are submitted to an Application Server.

Doc Share Request
A request created to post a vendor, customer, or project document to a SharePoint site using the Doc Share feature, which is defined and enabled in System Manager and Shared Information. A Doc Share request can originate from:
- Accounts Receivable Invoice and Memo (08.760.00)
- Accounts Receivable Statements (08.600.00)
- Order Management Invoice (40.680.00)
- Order Management Order Confirmation (40.610.00)
- Order Management Shipping Notice (40.654.00)
- Purchasing Print Purchase Orders (04.600.00)
- Flexible Billings Invoice Print (BI.INV.00)
- Flexible Billings Construction Billing Print (BI.CNP.00)

Polling Interval
How often, in seconds, the Application Server should poll the Application Server queue for new requests to run.

Process Request
A request originating from a process screen such as General Ledger module’s Release General Ledger Batches (01.400.00).

Quick Send Request
A request created when a document is sent to a vendor, customer, or employee via email or fax using the Quick Send functionality in the Shared Information module. A Quick Send request can originate from:
- Purchasing Print Purchase Orders (04.600.00)
- Accounts Receivable Invoice and Memo (08.760.00)
- Accounts Receivable Statements (08.600.00)
- Order Management Invoice (40.680.00)
- Order Management Order Confirmation (40.610.00)
- Order Management Manual Order Confirmation (40.611.00)
- Order Management Shipping Notice (40.654.00)
- Flexible Billings Invoice Print (BI.INV.00)
- Flexible Billings Construction Billing Print (BI.CNP.00)
- Service Dispatch Service Call Invoice (SD.640.00)
• Service Dispatch Reprint Service Call Invoice (SD.650.00)
• Service Contracts Service Contract Invoice (SN.640.00)
• Service Contracts Reprint Service Contract Invoice (SN.650.00)
• Payroll Direct Deposit Advice Slips (02.635.00)

**Report Request**

A request originating from a report screen such as the Inventory module's *Inventory Valuation* (10.620.00) report.
Index

A

Application Server
  changing the polling interval  22
  checking the queue  15
  deleting requests  35
  hardware environments  2
  launching  14
  operating sequence  2
  request processing parameters  43
  request status  14
  shutting down  19
  submitting requests  27, 36

B

buttons
  Begin Processing  27
  Print  27

C

changing the polling interval  22
checking the Application Server queue  15

D

deleting requests  35

E

email
  addresses  53
  processing requests  10, 71
  profile  6, 70
  setting up  6
  submitting requests  32

L

launching an Application Server  14

P

polling interval, changing the  22
Print button  27
processing
  email requests  10

Q

queue
  checking  15
  deleting requests  35
  polling  14
  polling interval  22

R

requests
  deleting  35
  Doc Share  1, 25
  process  25
  processing parameters  43
  Quick Send  1, 10, 25, 70
  report  26
  submitting  27, 36
  Transaction Import  26, 27, 33
  types of  25

S

screens
  Application Server Administration (96.080.00)  22
  shutting down the Application Server  19
  status of requests  14, 56
  submitting
    requests  27, 36
    Transaction Import requests  33

T

Transaction Import
  options  31, 33
  requests  27