

TechGuidePDF

Table Of Contents

Guidelines for Installation and Configuration of Agresso Business World	1
INTRODUCTION	1
Content overview	1
Conventions	1
What's New?	2
WHAT'S NEW IN TECHNICAL GUIDELINES	2
RELEASE HISTORY - TECHNICAL GUIDELINES	3
Installing Agresso	4
INSTALLATION OPTIONS	4
INSTALLING AND USING THE AGRESSO DEMO SYSTEM	5
Complete Agresso Installation	7
Configuring Agresso	28
Agresso Management Console	28
Server Configuration	65
Client Configuration	94
Software Distribution Server	107
The Agresso Self Service client	117
Agresso Web Services	126
General System Configuration	128
Upgrading Agresso	148
From Agresso 5.6.0 to Agresso 5.6.1	148
From Agresso 5.5 SPx to Agresso 5.6	160
From Agresso 5.4 to Agresso 5.6	172
Appendix	252
TABLES USED BY AMC	252
A NOTE ON CRYSTAL REPORTS	253
CONDITIONAL FUNCTIONS: CASE	262
SAMPLE	263
56 News	266
ACCESS TO ORACLE FILES	266
AGRESSO IN A TERMINAL SERVER (CITRIX) ENVIRONMENT	267
SDS AND WINDOWS FIREWALL (CLIENT)	268
SDS AND WINDOWS FIREWALL (SERVER)	269
ORACLE SYNTAX FOR CORRELATED UPDATES	271
Typical Agresso network topology	272
CERTIFICATE REVOCATION CHECK	273
Agresso Database Tools	276
Visual Basic for Application (VBA)	313
Security	320
Supported Platform - Agresso Business World Route 66	325
AUTHENTICATION	325

ABW56 - Technical Guidelines



Agresso Technical Guidelines

Guidelines for Installation and Configuration of Agresso Business World

You will always find the latest version at <http://abwdocs.agresso.no>

- [What's New in Technical Guidelines](#)
 - [Agresso Installation](#)
 - [Configuring Agresso](#)
 - [Upgrading Agresso](#)
 - [Agresso Maintenance](#)
 - [Security](#)
-

Agresso Business World Route 66 Milestone 1 (5.6.1)
[Copyright © 2011 Agresso R&D AS. All Rights Reserved.](#)

INTRODUCTION

Content overview

These Technical Guidelines give you detailed information on how to install and configure Agresso Business World. The content is divided into the following main sections:

- **What's New?** Explains briefly the new aspects with these Technical guidelines, compared to previous versions.
- **Installing Agresso.** Describes in detail how you carry out a complete Agresso installation, including database installation, client installation and server installation. You will also find information about how you can install a stand-alone demo version of Agresso on a single computer.
- **Configuring Agresso.** Describes how you use the *Agresso Management Console* to configure Agresso according to your company's needs.
- **Upgrading Agresso.** Describes the necessary steps to upgrade a previous database installation to the current version.
- **Agresso Maintenance.** Describes necessary operating (day-to-day) routines for your database installation, as well as important stuff about tuning and security.
- **Appendix.** This section covers technical subjects and goes more into detail regarding Agresso Database Tools.

Conventions

Symbols used in this documentation:

-  *Important information written in Italic*
-  Tip
-  Hide/Show a screen shot
-  Hide/Show a table
-  Question in FAQ
-  Drop-down text that is hidden until clicked

Anything written in this document is in principle not legally binding in declaration of intent.

What's New?

WHAT'S NEW IN TECHNICAL GUIDELINES

Release and update information

Current document release:	1.0 (Route 66 Milestone 1, Platform 2.5)
Release date:	2011-05-13
Last updated:	View update history

Read the [release history](#) for a list of the latest modifications to technical guidelines.

Notes on the Route 66 Milestone 1, version 1.0 release of Technical Guidelines



Plug and play upgrade scripts are not distributed!

Currently, the routines described in the topic [Copy setup between databases](#) shall NOT be used. However, **Unit4 R&D** has a series of scripts that can be used by experienced consultants.

Please send a request to the official **R&D** forum if you are searching for particular upgrade routines.

.Net 4.0 is required!

Agresso Business World Route 66 is based on Agresso Platform 2.5 and requires [.Net 4.0 Full Profile](#). All users of the **ABW Smart Client** must have .NET 4.0 Full Profile installed on their local computer!

Important - running Smart Client from server: Users that run the Smart Client from a server (Centrally Configured Client) may find that the Smart Client simply will not execute, with no warning.

This will happen if there is no (older) version of .NET Framework installed on the local computer, which may be the case for computers with relatively new Windows versions.

New functionality

ABW 5.6 contains new functionality (new solutions) for the following areas covered by Agresso Technical Guidelines:

- **Logging:** The new solution for logging is detailed described in [Logging in Agresso](#) in the Appendix. Details about specific setup via **AMC** is found in [Configuring Logging](#).
- **Process Execution Control:** You can now monitor all server queues and write status information to Windows event log.
- **ASQL:** A series of new ASQL functions are made available in ABW 5.6. See [Data Types and Functions](#) and [Data Manipulation Language \(DML\)](#) in the Appendix. For all ASQL update details, see (separate PDF-document) *Release Notes, Agresso Platform 2.2, News Introduced in the PLRA Project*.

- A New environment variable is added EXPAND_PATH_WITH_CLIENT. See [Initialize and Maintaining the Business Server Environments](#).
- You can now set Self Service in maintenance mode from AMC, meaning that a user will get a friendly message when the self service client is taken down. See [Managing Agresso Self Service](#) from more details.
- [Agresso Event Server](#) replaces the Agesso Alert Server web service.

Updates between release dates

The Technical Guidelines allows immediate updates and corrections. By publishing the updates to our web site, the last and complete version will always be available, for all reader groups. For all important releases, the decimal in the version number - starting at 1.0 for each major release and service pack - will be increased.

You will get all update details by following the link [View update history](#) at the top of this page.

RELEASE HISTORY - TECHNICAL GUIDELINES

The table lists all published versions of the Technical Guidelines, with last version on top:

Version	Technical Guidelines	Comments
5.6.1	2011-12-15	Oracle 11.2 workaround no longer needed from Oracle 11.2.3.0. Also updated Finalise Upgrade documentation.
5.6.1	2011-10-25	Info on oracle 11.2 patches required by Agresso. See Agresso on Oracle .
5.6.1	2011-06-30	Removed information about "show content pane" in the web help documentation , since this options is no longer available in Agresso 5.6
5.6.1	2011-06-06	Updated Oracle update samples with a sample for merge into syntax on oracle 11.2
5.6.1	2011-05-31	New database setup requirements for MS Sql Server and Oracle , to allow the database to read external files. This is needed after applying the first Software Update for Route 66 Milestone 1. (mssql: "grant administer bulk operations to <login>", Oracle: "grant create any directory")
5.6.1	2011-05-13	Agresso Business World Route 66 Milestone 1
5.6.0	1.1, 2011-05-10	Updated security guidelines .
5.6.0	1.0, 2010-10-30	Important improvements on server.
5.5.3	1.1,2010-05-21	Updated the SDS and Windows Firewall(client) documentation.
5.5.3	1.1, 2010-07-05	Added link to latest report engine installation on the Report Engine Installation page
5.5.3	1.1, 2010-06-05	Changes to Agresso On MS SQL Server. Updated list of things to remember when creating an Agresso database on MSSQL. "view server state" must be granted.
5.5.3	1.1, 2010-03-11	Changes to Agresso On MS SQL Server. Updated recommendation for Agresso on MSSQL. It's recommended to set Page verify to CHECKSUM.
5.5.3	1.1, 2010-01-06	Added documentation on how to configure access for SDS in the windows firewall on the server.
5.5.3	1.1, 2009-09-14	Changes to Agresso On MS SQL Server. Updated recommendation for Agresso on MSSQL (READ_COMMITTED_SNAPSHOT on). And made it clear that agrtempdb is optional.

5.5.3	1.1, 2009-04-18	ASQL documentation updated. New functions and datatypes documented, and more samples added
5.5.3	1.1, 2009-03-05	Note about agr_session view on an Oracle RAC
5.5.3	1.1, 2009-02-04	Minor corrections on database installation.
5.5.3	1.0, 2008-09-15	Built on top of Agresso Platform 1.1
5.5 SP2	1.1, 2008-04-10	Added environment variables. All available Agresso environment variables are now documented and indexed.
5.5 SP2 Update 0.2	1.1, 2007-09-14	Improved the upgrade process from 5.4 to 5.5.2
5.5 SP2	1.0, 2007-06-01	First version for ABW 5.5 Service Pack 2
5.5 SP1	1.1, 2006-03-13	Added printable PDF version of TechGuide55. Minor corrections.
5.5 SP1	1.0, 2006-02-01	First final version.
5.5	2005-07-14	First BETA version is html format.

Installing Agresso

INSTALLATION OPTIONS

Complete installation versus Demo installation

Complete installation

The main topics in this section cover a complete ABW installation in a complex environment, where you will need to install all of the following:

- Database tables
- Server components
- Client components
- Help system

Most parts of the ABW system will be installed by the **Agresso Business World** installation wizard. The exceptions are Agresso Report Engine and Agresso Web Help. These components must be installed separately.

Note: Please note that these guidelines do not cover the installation of a database system, i.e. **Oracle** or **MS SQL**.

The Installation wizard

Installation options

The file **SetupABW.exe** on the installation DVD, contains wizards for installation of all the Agresso components. Normally, the DVD will boot automatically.

The main installation options are explained below:

Option	Description
Agresso Business World	Allows you to install one of the following <ul style="list-style-type: none"> • Complete ABW. You can de-select unwanted components. • Smart Client. Installs the Smart Client component only. The wizard has de-selected the other components for you. • Demo. Installs the Agresso demonstration version on a single computer.
Agresso Demo	Runs scripts for configuring Agresso for demonstration purposes. This option requires that Agresso Business World is installed. The Agresso Demo setup scripts will be executed from the Agresso Business World installation when the "Agresso Demo" installation options is selected.
Web Help	Installs the Web help system on your web server. There is one installation for Smart Client and one for Self Service.
Bar Code	Installs the Agresso Bar code reader.
Agresso OCR	Installs optical character recognition software used by document archive.

INSTALLING AND USING THE AGRESSO DEMO SYSTEM

Purpose

The Agresso Demo Setup option allows the user to automatically configure Agresso on a single computer for demonstration purposes. The Demo system will automatically install MS SQL Server 2008R2 Express if no supported local Sql Server instances exist.

Install the Demo system

1. Start the [Agresso Business World](#) installation from SetupABW and select the [Agresso Demo](#) option.
2. Follow the instructions on screen until the installation is complete.
3. Click [Finish](#) to start the Agresso Demo Setup scripts.

The Agresso Demo setup scripts can also be launched on an existing complete Agresso Business World installation by using the Agresso Demo option available in SetupABW.exe. The Agresso Demo Setup scripts can be executed as many times as you want, but keep in mind that existing configurations for Self Service and Web Services (web.config) will be overwritten when running the setup scripts.

You can log in to the Agresso demo system as user [sysenlong](#) and with password [agresso](#).

Installed components

Overview

The following Agresso components will be configured by the Agresso Demo Setup scripts

- Business Server
- Self Service
- Web Services host
- Smart Client

Database server details

The database server settings are by default as follows:

Item	Value
MS SQL Server instance name	AGRDEMO561
Database name	agrdemo561
Database login	agrdemo561
Database login password	agresso
Sa password	AgrDemo561
ODBC datasource name	agrdemo561

The MS SQL Server instance:

When connecting to an SQL Server instance, the computer name be added as a prefix. If the instance name is AGRDEMO561, the instance name to connect to, is therefore:

<machine name>\AGRDEMO561

Using the demo installation

Installed features

The Demo system gives you full access to the ABW standard clients (Smart client - windows application, and Self Service client - Web application), using `systenlong` as username and `agresso` as password.

The Agresso database contains so-called template data, allowing you to test and demonstrate most of the ABW functionality.

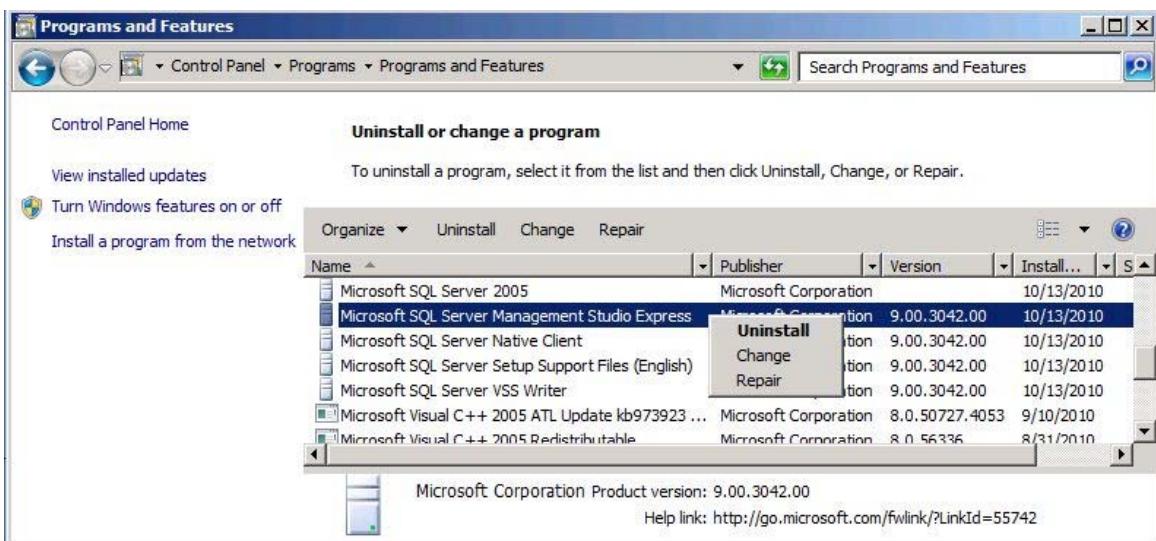
Requirements

- Complete Agresso Business World installation
- Windows powershell v2.0 (Windows Powershell is installed by default from Windows Server 2008 R2 and Windows 7)
- 64-bit Operation system, Windows Vista (6.0) or higher.
- [SQL Server 2008 requirements](#)

Powershell v2.0 will automatically be installed if the Agresso Demo Setup scripts are started directly from SetupABW. Windows Powershell v2.0 can also be downloaded from:
<http://support.microsoft.com/kb/968929/en-us>.

SQL Server 2005 Express Tools must be uninstalled

You will be unable to install SQL Server 2008 if the SQL Server 2005 Express Tools are installed (included with the 553 Agresso Demo installation). You need to uninstall this feature to be able to install SQL Server 2008. To uninstall go to Programs and Features in Control Panel and select Uninstall on the "Microsoft SQL Server Management Studio Express" installation. SQL Server 2008 Management Studio can also be used for SQL Server 2005 instances.



Logging (for troubleshooting purposes)

The table below shows where you find the various log files when installing Agresso Demo.

Type	Location
AgrDemo installation log-file	%TEMP%\AgrDemoSetup561\
SQL Server 2008R2 Express installation log	C:\Program Files\Microsoft SQL Server\100\Setup Bootstrap\Log

Complete Agresso Installation

THE INSTALLATION PROCESS

Prerequisites

Supported database platforms

Agresso requires either an Oracle or an MSSQL database system.
See [currently supported platforms](#).

Sufficient access rights for installer

The account used during the installation must have sufficient rights to be able to write to the registry during the installation, both to HKEY_CURRENT_USER and HKEY_LOCAL_MACHINE. The account must also have sufficient rights to install files to the system32 directory.

Process description

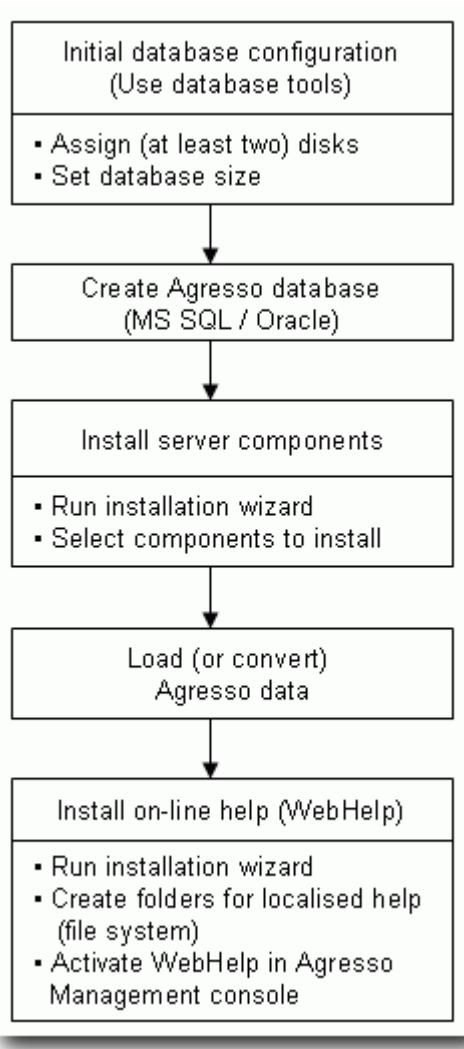
When you have the RDBMS up and running, the database must be configured according to the Agresso requirements. More than one thousand Agresso tables are in effective use, and needs to be taken care of.

Next, you must install the server components. Among the components you find **Agresso Management Console**, which is the main tool for later configuration tasks.

Finally, you complete the Agresso installation by installing the on-line help system.

Process diagram

The diagram below outlines the main activities in installation process:



Database Installation and Initial Configuration

Data File Distribution

INITIAL CONFIGURATION

Minimum requirements

The Agresso database must have at least two disks available, one for data and one for log. It is possible to install it on a single disk, but this is only recommended if your installation is very small, or it is merely a test installation.

All update queries sent to the database causes write activity to one or more data device pages and to the log device. If both data and log are stored on the same disk, these two writing processes will interfere with each other, which may result in poor performance.

Recommended hardware setup

Storage Area Network (SAN)

SAN can be used for the database and the Agresso servers.

Standalone disks and RAID (Redundant Array of Independent/Inexpensive Disks)

The number of separate disks for data and log storage is one of the most important factors determining the performance of your Agresso system. Generally speaking, the more disks, the better performance.

The temporary tables in Agresso should also have a separate disk. If not, some of the larger server processes in Agresso may be very slow for large volumes of data.

The number of disks and the number of disk controllers determine the efficiency of your data distribution. How many disks you need depends on the size of your database, the number of concurrent Agresso users, the profile of these users (batch, on-line processing), etc. It is possible to use 7 or 8, maybe 10 disks for Agresso and your RDBMS combined. In most installations this is not feasible, and probably not a good idea anyway. Between 3 to 6 disks should be sufficient. This depends on the size of the database and the number of concurrent users.

Disk controllers

The number of disk controllers depends on the number of disks and on the speed and capacity of each controller. Generally, in a multi-disk environment, more disk controllers will speed up disk access, and thus speed up Agresso. If you install extra disks on a computer, it may be a good idea to add an extra controller. Your hardware vendor may help you determine the number of controllers necessary for your disk configuration.

RAID and disk mirroring

In general, Agresso recommends using several separate disks to speed up the application. If using RAID, make sure that at least two disks are available outside the RAID system. The database log and the temporary tables in Agresso need to have their own separate disks since both log and temp tables involve sequential write operations. For security reasons you may also want to use a separate disk for mirroring the log files. Mirroring of the disk containing the Agresso temporary tables is not required.

! Remember that RAID is not 100% safe. Thus archiving and backup is still essential.

The type of RAID to use depends on your priorities, security, the amount of available disk space and performance. RAID 1 is simple and gives maximum security (complete mirroring of all files), but it requires a lot of disk space and involves a lot of writing activity. A RAID controller with a lot of memory cache may help speed up the use of RAID.

AGRESSO DATA SEGMENTS

Five segments for the Agresso data objects

The data objects in Agresso can be distributed among five different segments:

agrdata	Contains the Agresso tables without BLOB/IMAGE columns.
agrblob	Contains all the tables with BLOB/IMAGE.
agrdoc	Contains only the adsfileblob table (used in document archive).
agrindex	Contains the Agresso indexes.
agrscratch/agrtempdb	Contains real tables used for temporary data during the execution of server processes.
temp/agrtemp/tempdb	Contains real temporary tables used for temporary data during the execution of server processes.

Device versus segment

In some databases there are only one relevant level of data distribution. For instance, Oracle puts user tables in tablespaces. Each Agresso segment is equivalent to one tablespace. All default tablespaces in the Create Database Scripts delivered by Agresso should be created in every installation.

If you only have a few disks, there is no real point in creating a physical device for each Agresso segment (in these RDBMS's). You may create one device on each disk and optionally add multiple segments on each device. **Agresso Copy**, used to load data files from Agresso into your database, uses the standard Agresso segments. Therefore, it is a good idea to create all these segments, even though you may have only two or three physical devices.

Number of Disks	2	3	4	5
RDBMS	1	1	1	1
LOG	2	2	2	2
AGRSCRATCH/AGRTEMP/AGRTEMPDB/TEMP	1	3	3	3

AGRINDEX	2	2	2	5
AGRDATA	1	1	4	4
AGRLOB	1	1	4	4
AGRDOC	1	1	4	4

The differences are outlined in the table:

	Oracle	MS SQL
RDBMS	The SYSTEM, UNDO, TEMP and TOOLS tablespaces. Moving the tablespaces UNDO and TEMP to other disks may affect performance in a large installation. Oracle control files may be placed on any disk.	The RDBMS Software and the MASTER device. Mirroring of master (for security) may be placed on any disk.
LOG	The redo-log files are normally placed on a RAID 1 (a set of three log files). If you wish to have mirroring of the log files, use another disk for the next set (MIRRORING). If you use archiving, the archive files may be placed on any disk (not a disk with database devices). However, the disk should have a lot of free space.	The log device contains the log for both the agresso and the agrtempdb databases .
AGRSCRATCH/AGRTEMP/AGRTEMPDB/TEMP	The tablespaces agrscratch, agrtemp and temp.	The data device for databases agrtempdb and tempdb.

AGRTEMP/AGRTEMPDB and TEMP/TEMPDB

Most server jobs (and reports) create one or more tables for temporary storing data that exist only for the duration of the job. These database tables are usually created at the beginning of the job and dropped at the end of the job.

Much of the total writing to the database in a typical Agresso installation is writing to these short-lived tables. Moreover, both in Oracle and SQL Server all updating of these are logged to the database transaction log.

For debugging purposes it is possible to skip the drop command on these tables by setting the parameter AGR_TMP_SAVE.

From Agresso 5.5 these tables are created as temporary tables. When using SQL Server this is done by giving the table name the following prefix: ##. When using Oracle, it is done by issuing a "create global temporary table" command. Oracle use the temporary tablespace assigned to the user, and SQL Server use TEMPDB for the instance to hold the real temporary tables.

Updates to temporary tables are not logged. This gives a significant speed improvement for both database systems.

A side effect is that some of the IO is redirected to the temporary tablespace for Oracle, and to the TEMPDB database for SQL Server.

Especially for Oracle installations the reduced logging is a major improvement considered the size of archive files to back up. Installations that do log shipping to a mirror will experience a significant reduction of data to transfer.

If AGR_TMP_SAVE is switched on, the tables will be created the same way as in Agresso 5.4.

If it is required to have the old type of tables, but not to keep them when the job is completed, the parameter AGR_USE_REAL_TABLES can be set to TRUE. This turns the behaviour back to the way it was in Agresso 5.4.

ABOUT DATABASE CONFIGURATION

Initial configuration of the database

Instead of giving a detailed description on how to install the Database software, Agresso recommends visiting database vendors web sites for downloading the necessary programs, patches and documentation. One advantage is that this documentation will always contain the latest information regarding installing, configuring, administering and tuning the database programs for Agresso.

There are three main steps for the installation of a database:

1. Pre-installation - information on how the OS should be installed/patched and all other requirements
2. Actual installation - information on how to install the software
3. Post-installation - information on how to configure the database components and how to get the patches

 We recommend reading the installation guide thoroughly and follow their recommendations closely.

 Agresso recommends installing the latest patch set release after a successful installation of the Database.

Agresso Management Console - your main configuration tool

When you later will configure the completed installation, you will mainly use the **Agresso Management Console** - a utility program that always must be installed on the Agresso server.

AGRESSO ON MS SQL SERVER

Reference

If you need help to install SQL Server we refer to: <http://www.microsoft.com/sql>

General SQL Server settings

Microsoft® SQL Server™ automatically tunes many of the server configuration options, thus requiring little, if any, tuning by a system administrator. Although these configuration options can be modified by the system administrator, it is generally recommended to keep the default values, allowing SQL Server to automatically tune itself based on run-time conditions.

However, if necessary, the following components can be configured to optimize server performance:

- SQL Server memory
- I/O subsystem
- Microsoft Windows NT® options

Create an empty Agresso Database on MS SQL Server

Below, we give a short overview of things to remember when you create an Agresso database.

- **Database:** You need a database for Agresso data (Agresso).
- **Collation:** We recommend `Latin1_General_CI_AS`, but supports all collations.
- **File group:** If you are going to use File group, please refer to [Agresso Data Segments](#).
- **Authentication:** Select `SQL Server Authentication`. Agresso does not support only Windows Authentication.

- **Language:** Select [English](#) as language. The language for the LOGIN has to be set to English. Agresso uses English date and time formats in the database. (This does not influence the date/time format you want to see in Agresso.)
- **DB owner:** You have to set the **DB owner** manually to your Agresso login. Start [SQL Server Management Studio](#) and run the [*sp_changdboowner*](#) on your Agresso database.
- **DB owner:** You have to give the DB owner access to a system view:
`grant view server state to <login>`
- **DB owner:** You need to give the DB owner access to administer bulk operations:
`grant administer bulk operations to <login>`
- **Default database:** You have to set default database (Agresso) for your Agresso login.

Database properties for Agresso database

We recommend:

1. Set the recovery Model to [Full](#).
2. Use the following settings:
 - Auto update statistics
 - Page verify = CHECKSUM
 - Auto create statistics
3. Set the parameter READ_COMMITTED_SNAPSHOT on by issuing the command:
`alter database <dbname> set READ_COMMITTED_SNAPSHOT on`

This parameter reduces or removes possible problems related to lock-waits and deadlocks.

Optional: AGR_TEMPDB

Creating a separate database for help tables is optional. If the AGR_TEMPDB [Agresso environment](#) variable is defined, the database specified will be used for help tables. If AGR_TEMPDB is not defined (default), then the Agresso database will be used for help tables.

Recommended settings for an agrtemp database:

1. Set the recovery Model to [Simple](#).
2. Use the following settings:
 - Auto update statistics
 - Torn page detection
 - Auto create statistics
3. Set the parameter READ_COMMITTED_SNAPSHOT on by issuing the command:
`alter database <dbname> set READ_COMMITTED_SNAPSHOT on`

This parameter reduces or removes possible problems related to lock-waits and deadlocks.

ODBC in a 64-bit Windows environment

Agresso supports the 64-bit version of Windows. Since Agresso Business Server and Agresso Smart Client is 32-bit, Agresso will run in the 32-bit windows subsystem (syswow64) and use the 32-bit part of the registry (wow6432node).

*Note: When creating the ODBC-datasource in a 64-bit environment, remember to start [odbcad32.exe](#) from the [windows\syswow64](#) directory and **not** the default one.*

AGRESSO ON ORACLE

Oracle Client in a 64-bit Windows environment

Agresso supports the 64-bit version of Windows. Since Agresso Business Server and Agresso Smart Client is 32-bit, Agresso will run in the 32-bit windows subsystem (syswow64). As a result of this, Agresso Business Server needs the 32-bit of the Oracle Client to run. 64-bit oracle client is required for Agresso Self Service and web services.

Note: 32-bit and 64-bit of the Oracle Client can be installed side by side.

Easy Connect

The easiest way to connect to oracle from Agresso is to use Oracle Easy Connect when creating an Agresso server or client data source. When Easy Connect is used, tnsnames.ora is not needed.

The connection string used for Oracle Easy Connect must be defined in the following format: <HOST NAME>:<PORT>/SID . You don't need to define port when the default port is used; normally the connection string format will just be: SERVER/SID.

To support use of easy connect, oracle must be configured to include EZCONNECT as naming method (HOSTNAME can also be used). Naming methods can be configured using Oracle Net Manager, or by editing sqlnet.ora manually. The NAMES.DIRECTORY_PATH section in sqlnet.ora must contain EZCONNECT to make easy connect work. If sqlnet.ora don't exist, easy connect will be used.

General Oracle settings

Although many configuration options can be modified by the DBA, it is generally recommended that these options are left with their default values, allowing Oracle to automatically tune itself based on run-time conditions.

If necessary, the following parameters can be configured to optimize performance:

- SGA - Under installation, only set the `sga_max_size` and `sga_target` and let Oracle automatically tune the SGA memory.
- CURSOR_SHARING - We have seen good performance benefits by setting set the `cursor_sharing` to `force` or similar (default: `exact`).

Unicode support

To support Unicode characters in the database, the following settings must also be in place:

- Character set must be set to `AL32UTF8`
- The NLS_LENGTH_SEMANTICS parameter must be set to `char`.

Requirement

AGR_SESSION view must be created.

You must create a view and a synonym manually in the SYS schema and grant select on those to the PUBLIC schema.

```

REM ****
REM *** Login as SYS to run this script
REM ****
create or replace view agr_session
as select audsid, terminal, process
from v$session;

create or replace public synonym agr_session for sys.agr_session;

grant select on agr_session to public;

```

! v\$session should be replaced with gv\$session when using Oracle RAC.

Create tablespaces and user

Prerequisites

Before you can load tables and data into an Oracle database, you have to:

- Create tablespaces for the Agresso data
- Create an Agresso user in the database
- Grant rights to the user

Procedure

Start SQL*PLUS and login as SYS/pwd as SYSDBA. Continue as follows:

Set default destination for all data files:

```
alter system set db_create_file_dest='C:\ora_data';
```

Create tablespaces:

```
create bigfile tablespace agrdata datafile size 100m autoextend on next 50m;
create bigfile tablespace agrindex datafile size 100m autoextend on next 50m;
create bigfile tablespace agrblob datafile size 100m autoextend on next 50m;
create bigfile tablespace agrdoc datafile size 100m autoextend on next 50m;
create bigfile tablespace agrscratch datafile size 100m autoextend on next
50m;
create bigfile temporary tablespace agrtemp tempfile size 100m autoextend on
next 50m;
```

! We recommend to set AGR_TMP_LOC environment variable in the Agresso Management Console to the agrscratch tablespace.

Create Agresso user:

```
create user agr56 identified by agresso
default tablespace agrdata
temporary tablespace agrtemp
quota unlimited on agrdata
quota unlimited on agrindex
quota unlimited on agrblob
quota unlimited on agrdoc
quota unlimited on agrscratch;
```

Create new ROLE for Agresso and grant session privileges:

```
create role agresso_role not identified;
grant create session to agresso_role;
grant alter session to agresso_role;
grant create table to agresso_role;
grant create trigger to agresso_role;
grant create view to agresso_role;
grant create procedure to agresso_role;
grant create sequence to agresso_role;
grant create synonym to agresso_role;
grant create any directory to agresso_role;
```

Grant privileges to Agresso user:

```
grant agresso_role to agr56;
```

Notes on Oracle 11.2

If you run Agresso on Oracle 11.2, you need to be aware that Oracle 11.2.0.1 has a serious error that will affect ABW. This problem has been solved in Oracle 11.2.0.2 and all Agresso Route 66 customers on Oracle 11.2 need to make sure that they are on Oracle patch level 11.2.0.2 or higher. Unfortunately Oracle patch level 11.2.0.2 also has an error that affects ABW. For this error there is a work-around and if you run on 11.2, you need to do the following:

Log in as sysdba and run this query (on Oracle 11.2.0.2):

```
alter system set "_optimizer_false_filter_pred_pullup" = FALSE scope=both;
```

In Oracle 11.2.0.3 this bug is fixed and it is not necessary to define the parameter.

You can find the oracle version you are running on by running the query:

```
select * from v$version
```

You are now ready to load the template database. Information on how this is done is described in the document [Loading a New Template Database](#)

ABW Installation

Full Installation

SERVER INSTALLATION OVERVIEW

One installation wizard

With the exception of [Web Help](#), all the **Agresso Business World** components are handled by one installation wizard. Still, you can install ABW components on separate servers.

Need for configuration

The Agresso Business World installation works only as a file copier. You will eventually use the **Agresso Management Console** (AMC) to set up and configure the Agresso Server according to your company's needs.

Available components

Although we recommend that you install all components, the **Custom Setup** window gives you the option to de-select certain components.



Custom Setup



The available main components are shortly described below

Component	Installation Directory	Description
Agresso Business World (32-bit)	<32-bit install location>\Bin	Contains the 32-bit Agresso components. Smart Client, Agresso Business Server and Agresso Management Console. Sub components...
Agresso Business World (64-bit)	<64-bit install location>\Bin	Contains the 64-bit Agresso components. Components required to configure Agresso Self Service and Agresso Web Services. 64-bit wmi providers are included to enable configuration of 64-bit components through AMC. Sub components...

64-bit and 32-bit components are installed to separate locations. By default Agresso Business World 32-bit components will be installed to `c:\Program Files (x86)\Agresso 5.6.1`, the 64-bit components will by default be installed to `c:\Program Files\Agresso 5.6.1`. You can customize these locations, but you are not able to change the sub-directories for the different components.

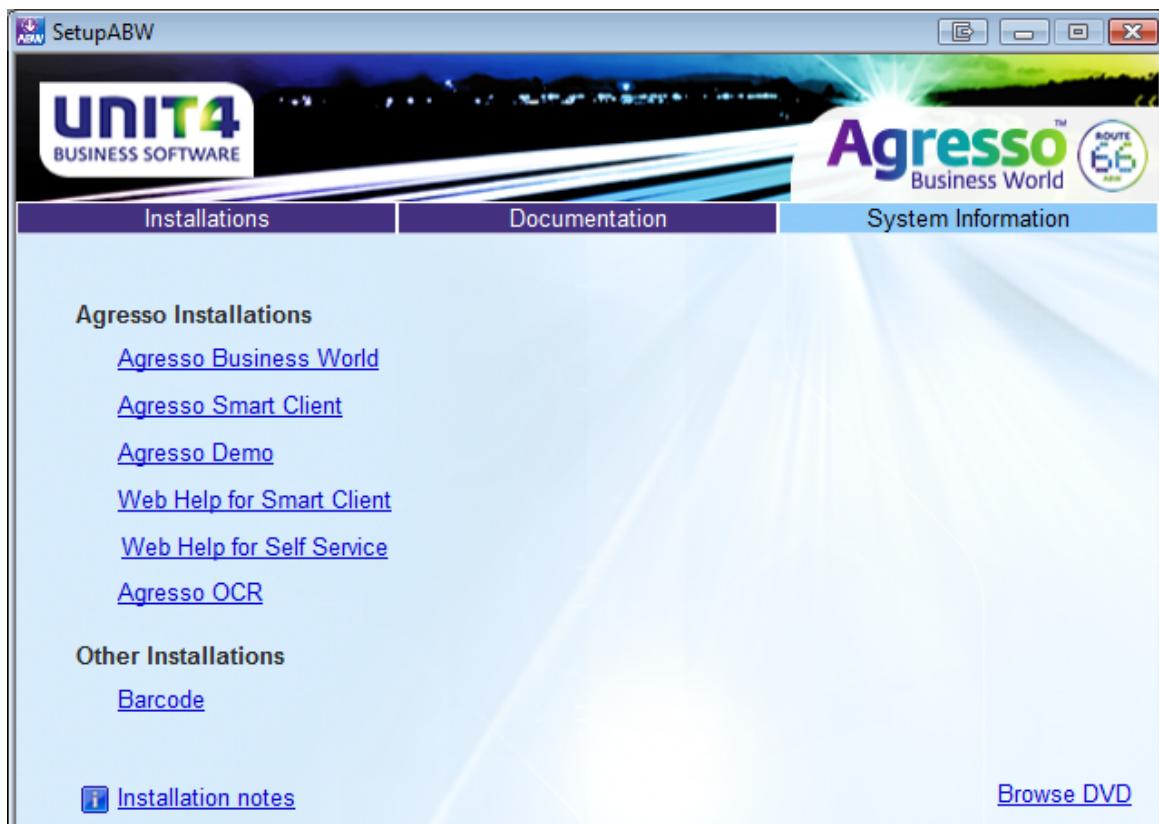
SERVER STEP-BY-STEP INSTALLATION

Detailed procedure

After booting the ABW Installation DVD, `SetupABW.exe` shall start automatically.

Note: If the DVD does not start, you can locate the DVD in **Windows Explorer** and double-click on `SetupABW.exe`.

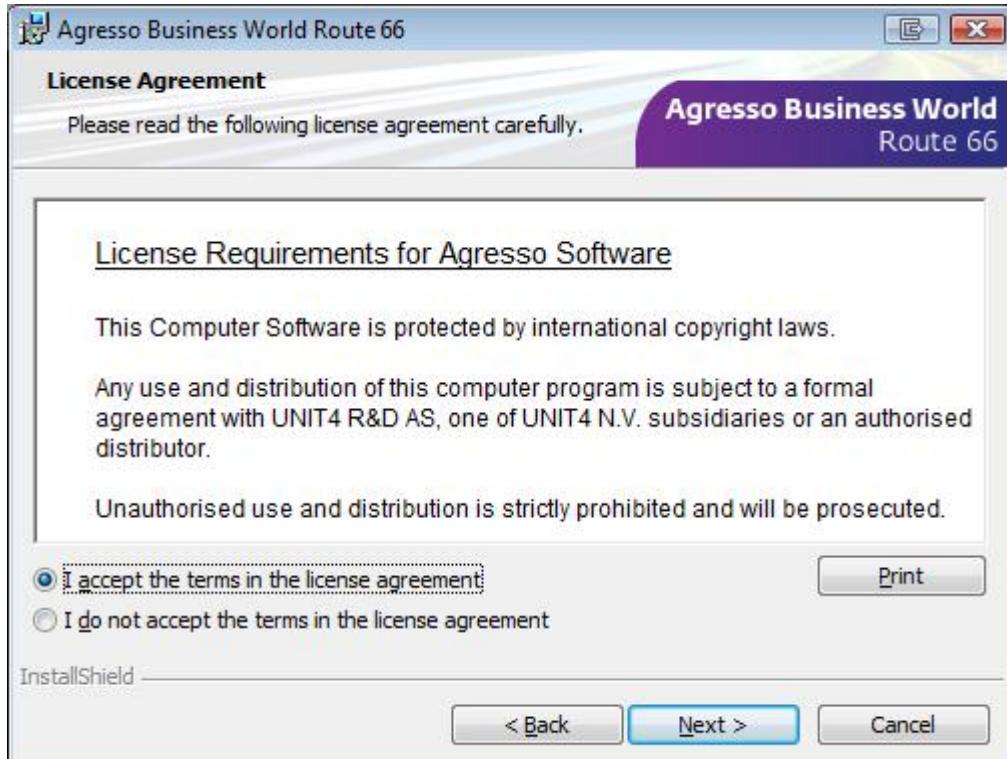
 Example: Setup ABW



Note: Please read the Installation note first.

1. Click on Agresso Business World.
2. Click **Next** until you come to the License Agreement window..

Example: License Agreement



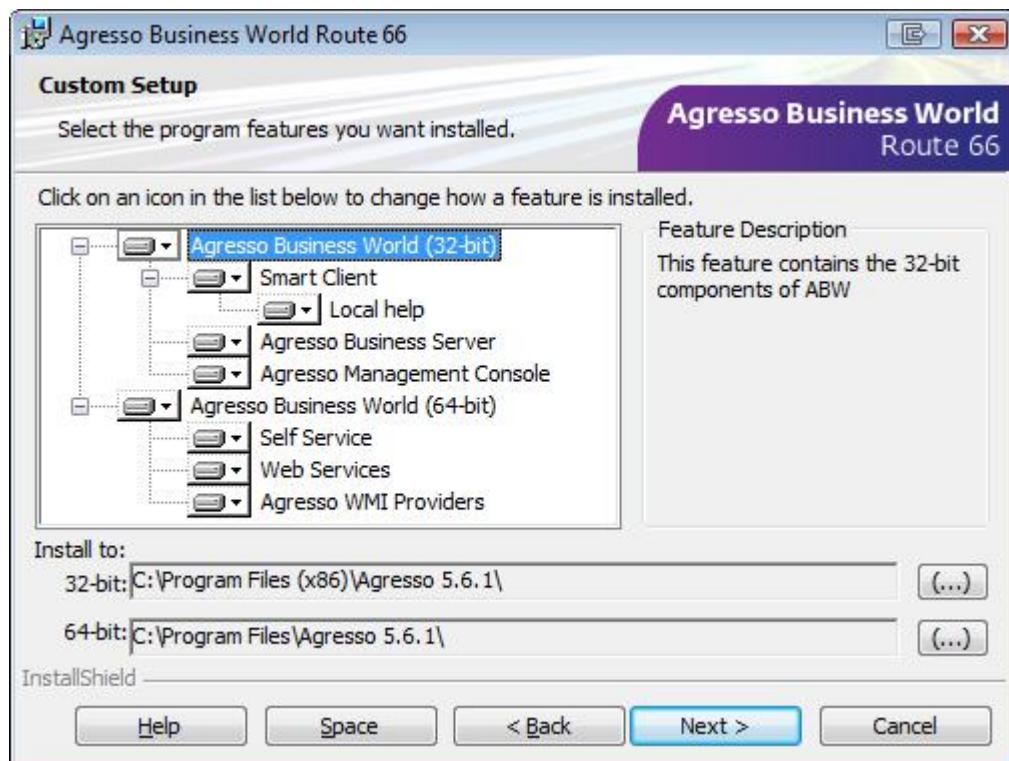
3. Make sure that you accept the terms in the license agreement, and click **Next**.
The program automatically extracts information like User Name and Organization from the computer.

4. Select the preferred installation option, and click **Next**.

The Setup Type window is displayed.

5. Select **Complete Agresso Business World**, and click **Next**.

 **Example: Server Installation**



For a standard server installation, you leave the settings as default.

6. Click **Next**.

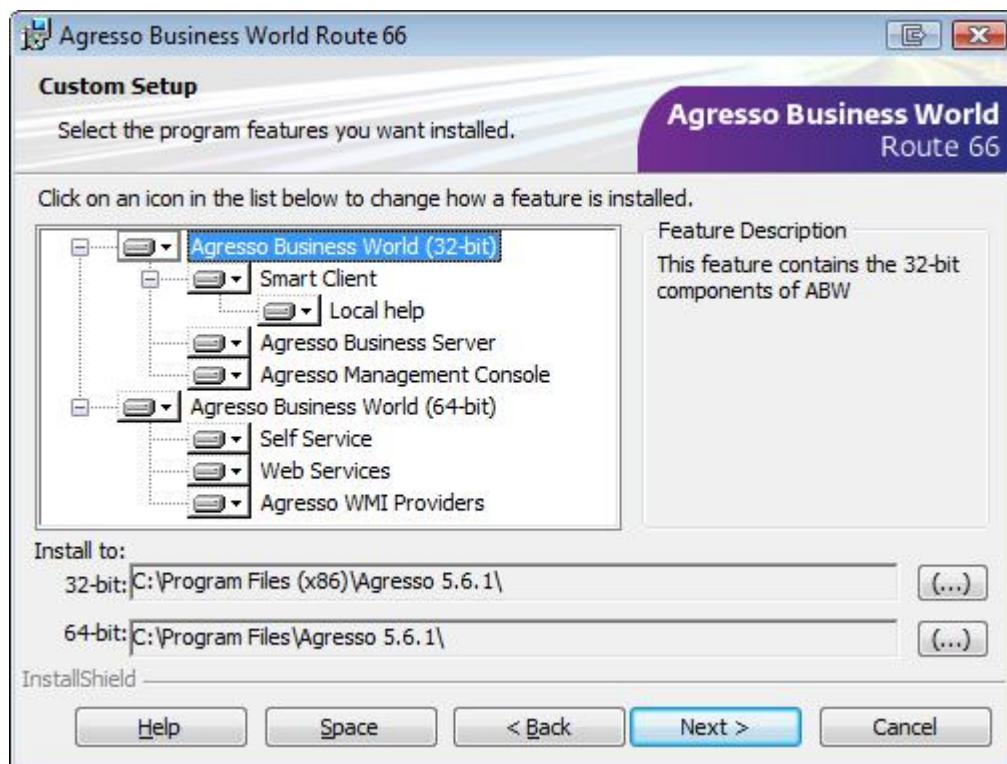
You are now ready to install the program.

7. Click **Install**.

The necessary files are copied. When completed, the installation is finished.

INSTALLING COMPONENTS

Available components



The available components are shortly described below:

Component	Installation Directory	Description
Smart Client	<32-bit install location>\Bin	Installs the files required to run the Agresso Smart Client. This component is also required when you are going to configure a centrally configured client .
Agresso Business Server	<32-bit install location>\Bin	Files required by the Agresso Business Server
Agresso Management Console	<32-bit install location>\Bin	Agresso Management Console AMC is required for most configuration tasks. If you need to configure 64-bit Agresso components like Self Service And Web services, you will also need to install the 64-bit WMI Providers. More..
Self Service	<64-bit install location>\Self Service	Files required to set up Agresso Self Service. More...
Web Services	<64-bit install location>\Web Services\	Files required to set up the Agresso Web Services. More...
Agresso WMI Providers	<64-bit install location>\bin	64-bit WMI Providers required to configure 64-bit Agresso components like Self Service and Web Services.

! On a 32-bit OS only the 32-bit components will be available. The Agresso server components requires Windows Server 2008 or later, and will be unavailable in the installation wizard on previous OS versions.

Note on Self Service and Web services

Application pool

You can install several instances of the Agresso Self service application, as long as each instance is installed in a separate *application pool*.

Installation description

General: The Self Service and **Agresso Management Console** components are required to setup and configure Ageresso Self Service. The Self Service component contains all the files necessary to setup and configure the Agresso Web Server using **Agresso Management Console**. See [Adding a Web Service](#) in the configuration section.

 *Configuring the web application manually without using the Management Console is not recommended and not supported. This also applies for the web services.*

ASP.NET Extension: Agresso Self Service and the Agresso Web Services use ASP.NET. ASP.NET web service extension is installed and enabled on the Internet Information Server when you set up Self Service or one of the web services.

.NET Framework: Installing .NET Framework will by default install ASP.NET.

Configuration

Please refer to [Adding a Web Service](#) in the configuration section.

Client Installation

SMART CLIENT

Smart Client installation only

When you need to install the Smart Client on local computers - or if you need to reinstall it on the server - you can select Smart Client Installation from the installation DVD.

Smart Client Installation will simply install the relevant client components from the Full installation option, without asking you to de-select all the other components.

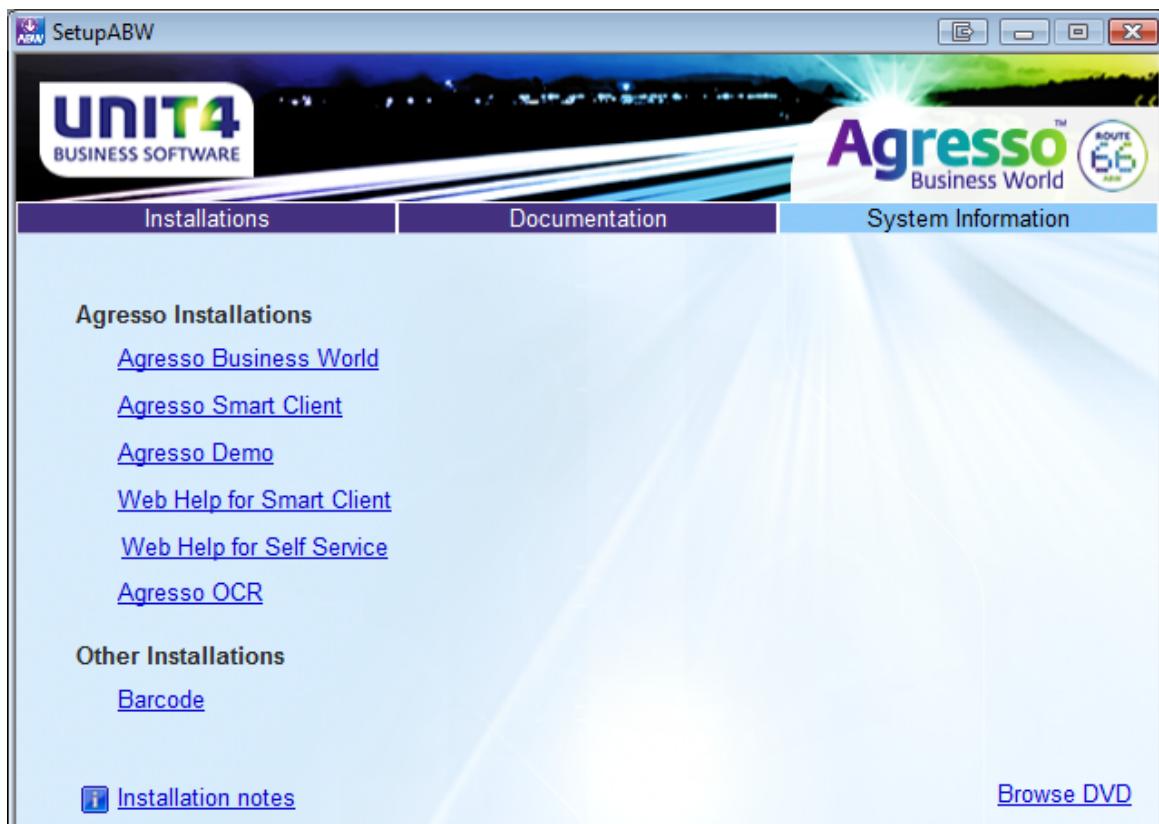
Server note: If you need to create centrally configured clients from the Agresso Management Console, you will need the Smart client on the server.

Step-by-step installation

After booting the ABW Installation DVD, [SetupABW.exe](#) shall start automatically.

Note: If the DVD does not start, you can locate the DVD in **Windows Explorer** and double-click on [SetupABW.exe](#).

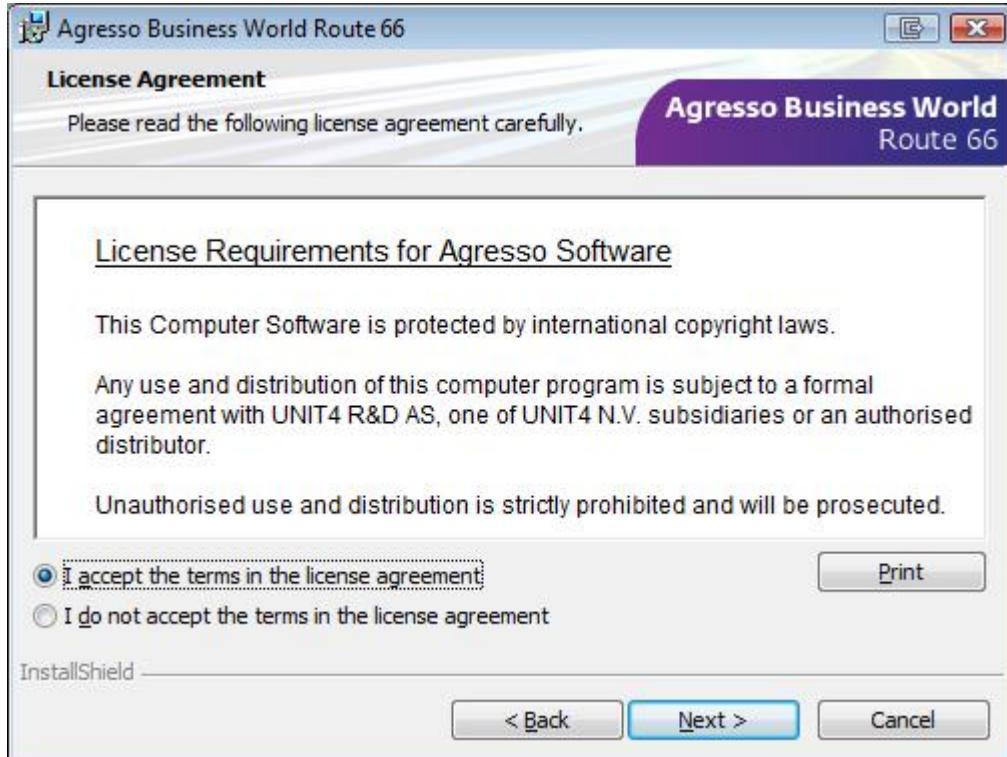
 Example: Setup ABW



Note: Please read the Installation note first.

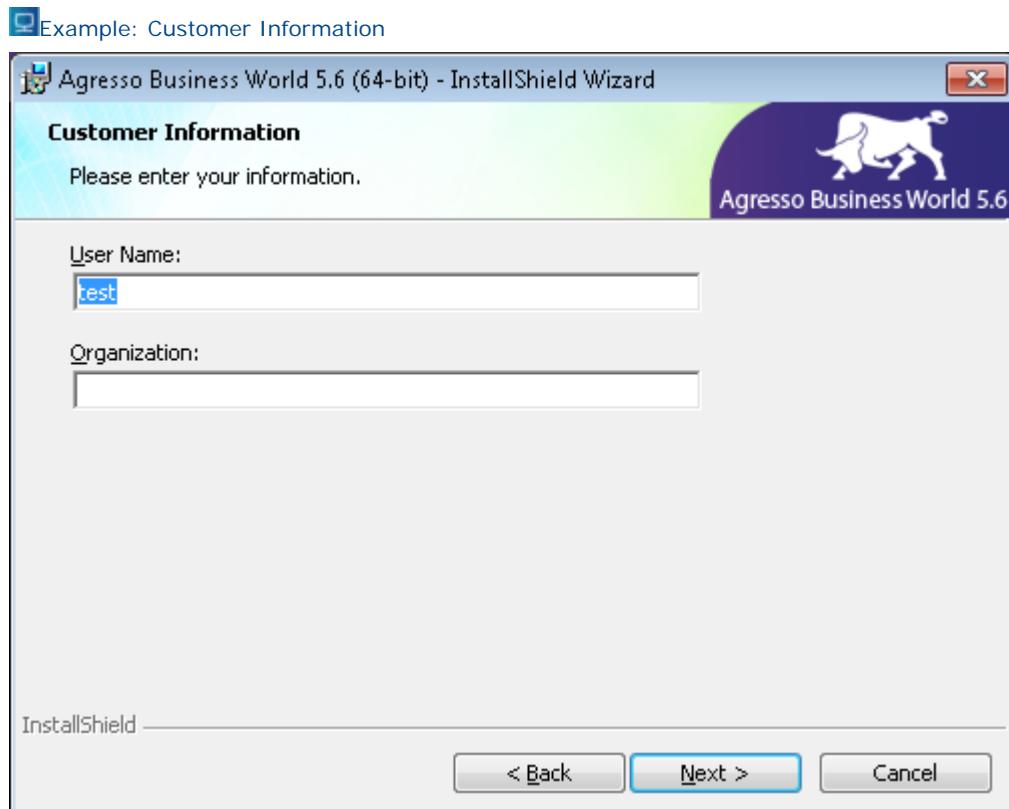
1. Click on [Agresso Business World](#).
2. Click **Next** until you come to the **License Agreement** window.

Example: License Agreement



3. Make sure that you accept the terms in the license agreement, and click **Next**.

The program automatically extracts information like **User Name** and **Organization** from the computer.



4. Select the preferred installation option, and click **Next**.

The **Setup Type** window is displayed.

5. Select Agresso Smart Client, and click **Next**.

The Smart Client is now ready to be installed.

6. Click **Install**.

The necessary files are copied to the computer and the installation is completed.

Related topics

[Data source setup](#)

[Agresso in a Terminal Server \(Citrix\) Environment](#)

LOADING A NEW TEMPLATE DATABASE

First time installation

General

When you install Agresso for the first time, you need to load the Agresso table definitions and template data into your database.

Some of the principles mentioned below, also apply if you need to copy Agresso data in or out of your database for other reasons.

Data distribution options

Storage spaces

There are approximately a thousand tables in Agresso Business World. To be able to distribute these tables to different disk locations, Agresso uses a set of logical storage spaces.

These will be created as *tablespaces* in Oracle, and *file groups* in SQL Server. For more information see [Agresso Data Segments](#).

Copy options

When copying data into Agresso, make sure the objects are stored in the correct data space. This requires using the options **-T** and **-I** with the Agresso Copy programs for tables and indexes respectively. [The Agresso Copy programs](#) are described in detail in the Appendix.

Copying of the Agresso tables with **-T** and **-I** options assumes that the Agresso standard data spaces already are created, and the tables and indexes will be correctly distributed.

Note: It is also possible to install all objects in a single logical space. This may be a good idea for a very small installation where data distribution is not essential. We do recommend, however, that you use the [Database Tools](#) in the [Agresso Management Console](#).

Load tables into the database

To be able to start the Agresso application, data must be loaded into the database.

The file *new.zip* is located in the Agresso database script directory: *.\Agresso 5.6\DatabaseScript\DbLoad*

Load procedure (1) using wizard

Do as follows:

1. Unzip *new.zip* (if not done already!).
2. Use the Database Copy utility from the [Agresso Management Console](#) (There is a log-in option on the right-click menu of the copy-in tool).

Load procedure (2) using copyora or copyms

Do as follows:

1. Unzip *new.zip* (if not done already!).
2. Copy the tables into the database by running the copy programs with the following parameters:

- **Oracle:**

```
copyora.exe -din -v -U<user> -P<password> -S<dbserver> "-f<path to folder containing files.lst>" -z -T -I -a5000 > copy.log
```

- **SQL Server:**

Unicode support: Note that the **u** flag must be set if your database shall support Unicode (shown as and optional **[-u]** in the command example below):

```
copyms.exe -din [-u] -v -z -U<user> -P<password> -Ddbname -S<datasource> "-f<path to folder containing files.lst>" -T -I > copy.log
```

After the command has completed, check the file *copy.log* for error messages. The copy programs is located in the bin-folder of the x86 installation.

Re-creating database views

To make sure all database views in Agresso are correct, you may want to re-create the views. Views can be re-created from the Copy In node in AMC. To re-create views, locate the Copy In node in under Database Tools, right-click and select [All Tasks | Recreate views](#).

[Database Tools | Database Copy | Copy In | All Tasks | Recreate views ...](#)

You may also re-create views using the copy tools from command line:

create an empty directory with an empty file called *files.lst*. In this new directory, run
copy<db>.exe -din <logon information> -v.

Create and re-create triggers and procedures

To make sure all database functions and procedures in Agresso are correct, you can re-create them. To re-create views, locate the Copy In node in under Database Tools, right-click and select Re-create triggers/procedures.

[Database Tools | Database Copy | Copy In | All Tasks | Re-create triggers/procedures...](#)

You can also re-create procedures from command line using the Agresso copy tools. To do this, you must create an empty directory with an empty file called *files.lst*. In this new directory, run

```
copy<db>.exe -din <logon information> -z
```

REPORT ENGINE INSTALLATION

It's recommended that the latest version of report engine is installed. The latest Agresso Report Engine version can be downloaded from the ABWUpdates page:

https://abwupdates.agresso.com/updates.aspx?product_area=REP

For updated information on how to install and configure the latest version of Agresso Report Engine, please refer to the Installation Notes document distributed with Report Engine.

Online Help Installation

WEB HELP INSTALLATION OVERVIEW

Options

The help system comes in two versions:

1. Web based help (WebHelp), where the help files (html-files) are located on a web server and uses a Web browser for display. There is one Web Help installation for Agresso Smart Client (SCHELP) and one for Agresso Self Service (SSAHELP). You can access these help files from many different Agresso 5.6 installations as long as a reference to the WebHelp server is added to the database through a configuration in [Agresso Management Console](#).
To install WebHelp, see [WebHelp Installation](#).
After installation, you need to publish the WebHelp system. See [WebHelp Configuration](#).
2. Compiled help (HTMLHelp), where the help files (the same html-files as for WebHelp) are compiled and packed into CHM files. These are installed on the network share during the ABW installation. The compiled help system is faster than Web based help, but can be accessed from the Smart Client only.
HTMLHelp does not require a specific installation, since the CHM files are copied to the correct directory during the ABW installation.
Configuration of the HTMLHelp is described in [HTMLHelp Configuration](#).

Language and installation folders

Language alternatives

Agresso offers the following language versions for Web Help:

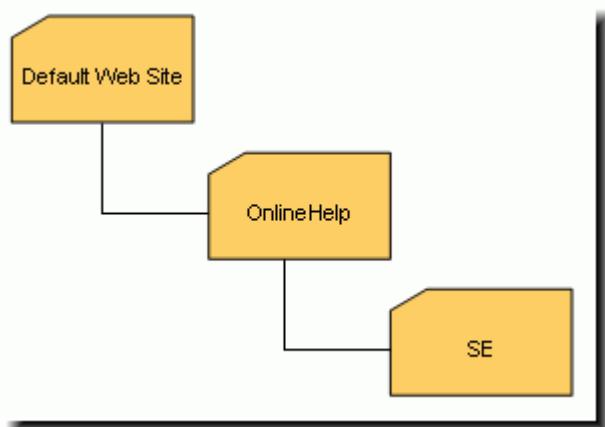
Installation folders

During installation, you must select the web site to use, and give a name for the Web Help entry point (web application).

If you have made these selections during installation:

- Language: `SE`
- Web site: `Default Web Site`
- Web application name: `SCHELP`

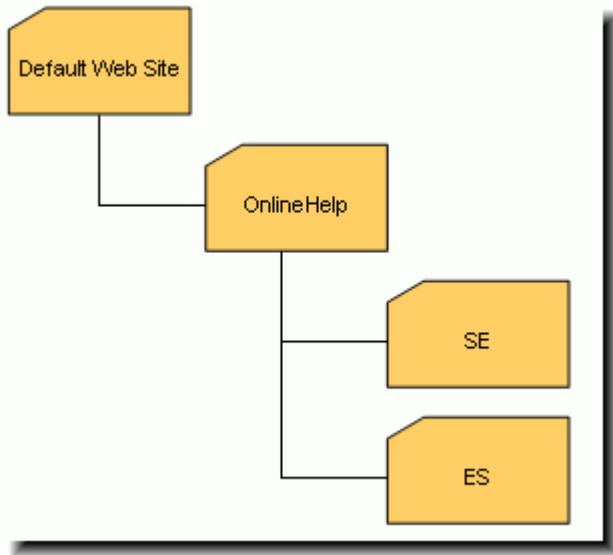
the following structure will be created in IIS:



The help files will be available in the SE folder.

Copying language folders

In cases where you must have several language codes available - with the same set of default help files - you can simply copy the folder containing the default files and rename it to the desired language code.



WEBHELP INSTALLATION

Wizard

You will use the [Web Help for Smart Client](#) and [Web Help for Self Service](#) link on the installation DVD to install the Help system.

Install WebHelp

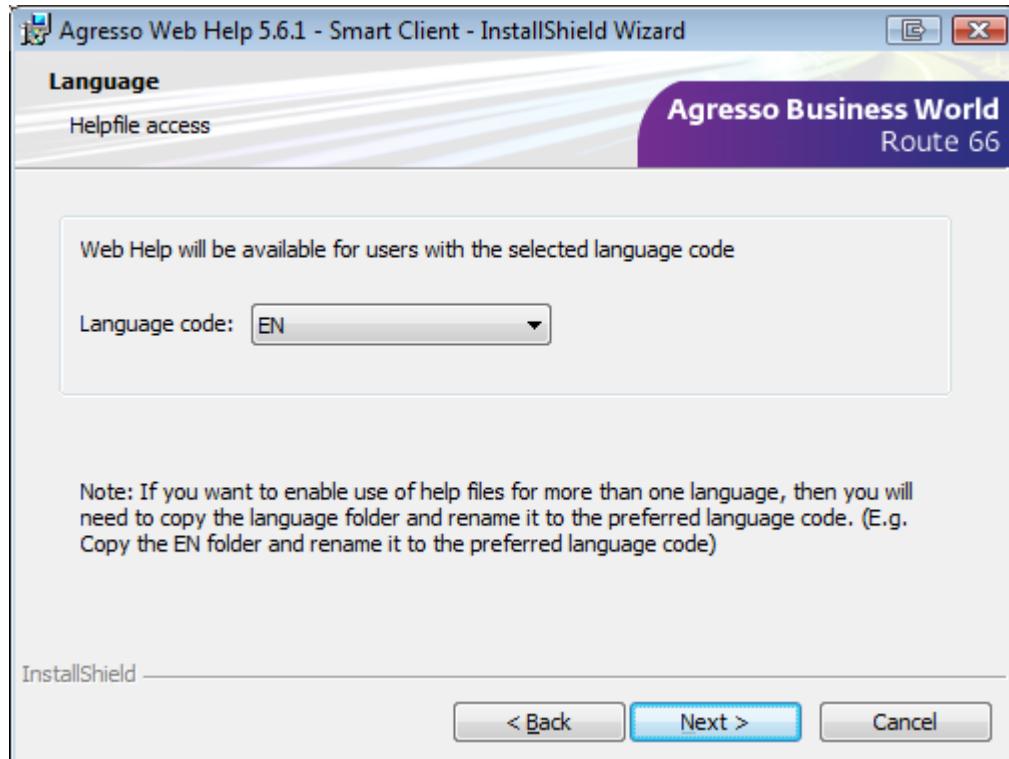
1. Click **Next** until you reach the **Customer Information** dialog in the installation.

2 A: Do as follows:

- Enter correct **User Name** and **Organisation** (or accept default values)
- Select **Anyone who uses this computer**
- Click **Next**

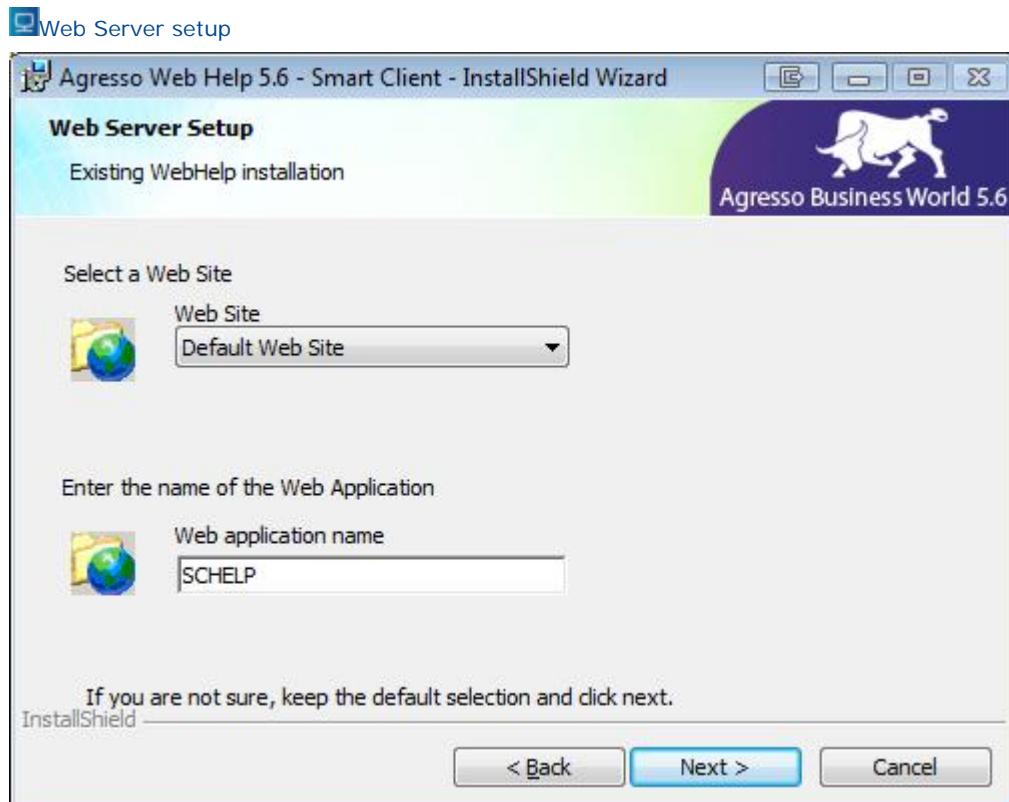
Note: If you selected default help language in step 1, you will now have to select the language code for the installation. The selected language code will give name to the folder where the help files will be installed:

 **Help Files Language**



3 B: Select language code and click **Next**

If this is your first installation, the **Web Server Setup** dialog will display. Otherwise, the already selected setup will be used, and you will only have to complete the installation - from step 6 below.



4. Select a **Web Site** and type in the name you want for the Online Help application. Then click **Next** to bring up the **Destination Folder** dialog.

5. Select the folder (in the file system) where you want your Online Help (root) catalog to reside, and click **Next**. The selected folder will be used as the entry point catalog for the language code folders in the Online Help installation.

6. Click **Install** to start the installation.

Configuration

When Agresso WebHelp is installed you must use the **Agresso Management Console** to configure the help system. See [Help Configuration](#).

Other deployment options

If you want extract the help files from the installation without installing the product, this can be done using the following command:

```
msiexec /a "<path to msi-file>" /qb TARGETDIR="C:\\"
```

In this sample the files will be extracted to `c:\program files\Agresso\Web Help\XX`. The XX folder must then be renamed to the correct language code, and a web application must be created manually if needed.

Advanced Installation Options

LOGGING AND TROUBLESHOOTING

Logging

Installation logging can be turned on by manipulating the following registry value:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer\Logging
```

Setting this value to `voicewarmup` will turn on verbose logging of the installation. Windows Installer log files are generated in the local temp directory of the user running the installation.

You can also specify logging as a parameter in the `msiexec`. The sample below installs Agresso 5.6 and creates a log file at a custom directory:

```
msiexec /I <path to msi package> /l*v c:\MyLogFile.log
```

If you experience problems with the Agresso Business World installation, please run the installation with verbose logging. The log file can be used for analyzing the problem.

SILENT INSTALLATION

Install from command prompt

[Windows Installer](#) has an option to run a silent installation.

A silent installation means running the installation in the background, with no user interface. All selections usually performed through the GUI during the installation process, must now be set from the command line.

Silent installation

Complete Agresso Business World

```
msiexec.exe /i "Agresso Business World (64-bit).msi" /quiet ALLUSERS=1
INSTALLLEVEL=150 INSTALLDIR="C:\Agresso_x64\" INSTALLDIRX86="C:\Agresso_x86\
/l*v %TEMP%\AgrABWCompleteInst.log
```

Smart Client

```
msiexec.exe /i "Agresso Business World\ABW x86 Setup\Agresso Business World
(32-bit).msi" /quiet ALLUSERS=1 INSTALLLEVEL=75 /l*v
%TEMP%\AgrSmartClientInst.log
```

Agresso Demo

```
C:\Windows\system32\WindowsPowerShell\v1.0\powershell -executionPolicy
Unrestricted StartAgrDemoSetup.ps1 -Silent $true
```

General syntax for silent installation

```
msiexec /i <path to msi package> /quiet
```

For more options see microsofts documentation of standard Windows Installer [commandline options](#).

Configuring Agresso

Agresso Management Console

AGRESSO MANAGEMENT CONSOLE

Installation Configuration and Maintenance

The **Agresso Management Console**, AMC, is the main tool for managing your Agresso installation. AMC complies with international standards for software management, and is based on

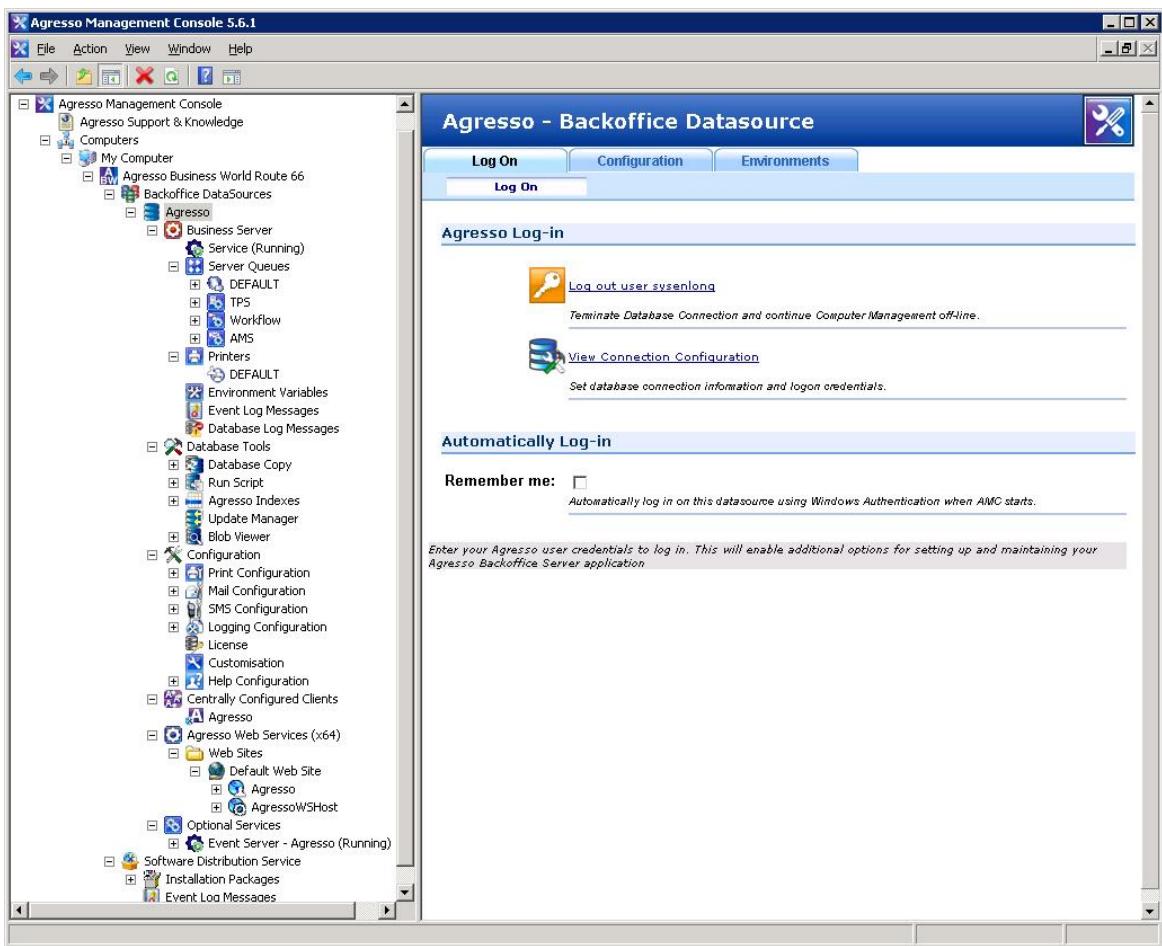
- **Microsoft Management Console** (MMC), which defines a standard user interface for managing applications, and
- **Windows Management Instrumentation** (WMI), which is a standard technology for accessing management information in an enterprise environment.

Agresso Management Console offers full local and remote set-up and management of your Agresso installation, with integrated security.

The management console is split in two panes,

- the *node pane*, and
- the *dialog pane*.

AMC example



Node pane

In the left pane you can navigate among servers and installed Agresso software. An item in the left pane is referred to as a **node**. All nodes have a context menu, activated when you right-click the node. Moreover, special actions available for each node are found under the Action toolbar menu item.

Some of the nodes available are based on what has been installed on the computer. Example: Agresso Server must be installed for the Business Server node to be available.

Dialog pane

The right pane will display a dialog for the selected node, where you can view and change the configuration settings.

Remote Administration

When the AMC has been installed on a server - during the Complete Agresso Installation process - you should also install AMC on your local computer, and from there enable remote administration of the server's AMC.

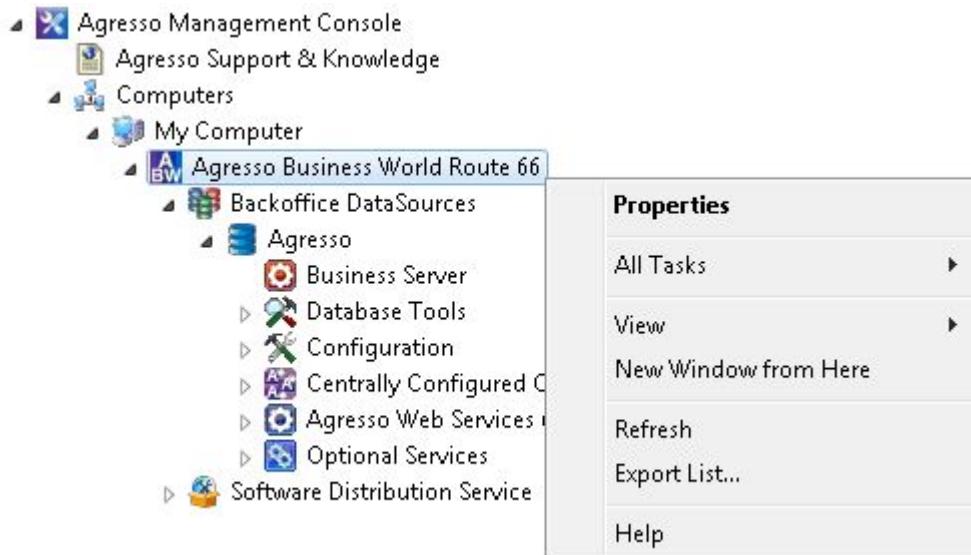
With the correct security settings, you can perform most of the Agresso configuration tasks without leaving your chair.

[Link: Enable Remote Administration](#)

AMC - GENERAL OPTIONS

General configuration

By right-clicking on the **Agresso Business World** node in **AMC**, you get access to some general properties, valid for the Agresso installation as a whole.



The following properties can be set:

Property	Description
Default max number of Events to display	Maximum number of events default shown for any EventLog node in the console. If you have performance problems when retrieving the events, try to lower this number.
Max number of Events to display for node	Allows you to set individual maximum number of events to be displayed for each EventLog node.
Process Views Update Frequency	The process views on the Business Server node and the Server Queue nodes are by default refreshed every 3rd second. This triggers a series of queries to be run against the database. If you notice performance issues caused by this, try to set a lower refresh rate.
Mail From the Console	Allows you to configure the settings used when sending mail from the AMC. This includes distributing the short-cut to an Agresso Centrally Configured Clients using mail.

CUSTOM WEB PAGE NODES

Customising AMC

New nodes

You can add your own custom nodes to access Agresso related information through an URL within AMC. The URL can be any valid URL used to access network resources, external system management and monitoring tools, files and folders.

Valid parent nodes

The following AMC nodes can be parents to Custom Web Page Nodes:

- Computer Node
- Agresso Version Node
- Data Source Node
- Business Server Environment Node

- Service Node
- Server Queue Node
- Customization
- Help Configuration
- Agresso Web Services
- Web Service Node
- Centrally Configured Client Node

Node properties

You set the node properties in the **Customized Node** dialog (see example below). The options are as follows:

Solve security issues

Remote administration through the **Agresso Management Console** is based on integrated security. You will need administrator rights to be able to do any changes to the remote computer just as you need administrator rights to do any changes to your local computer. This is governed by AMC's use of WMI.

Procedure

To set up remote access right to the WMI service, you must first open the **Computer Management Console** - on the computer (server) you want remote access to - console found in your PCs *Settings/Control Panel/Administrative Tools*.

1. Activate the **WMI Control Properties** window:

- a. Expand the **Services and Applications** node,
- b. right-click the **WMI Control node**,
- c. and select **Properties**.



2. Activate the **Security for Root** window:

- a. Activate the **Security property** tab,
- b. select the Root node,
- c. and click the **Security** button



3. Make sure that your user, or a group of users where you belong, has the **Remote Enable** permission set to **Allow**.
4. Save your work.
If no errors occurred, your access rights will now be in order.

Enable Remote Administration

Two tasks

To enable remote administration, you must

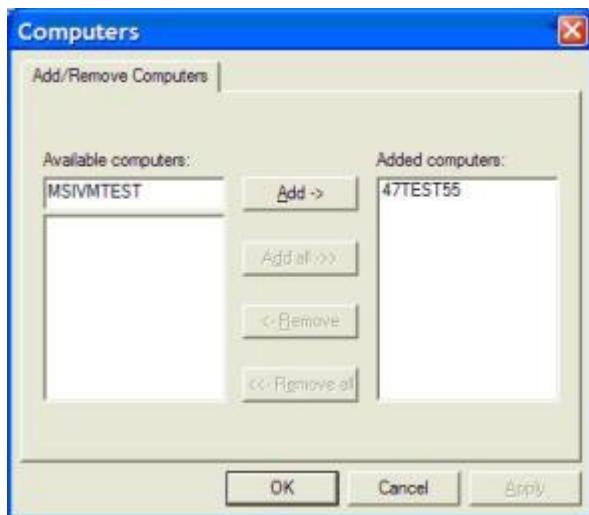
- 1) add the remote computer to your local AMC, then you must actively
- 2) connect to the remote computer.

Add the remote computer to AMC

To enable remote administration of your Agresso installation, you will need to add the remote computer (server) to your local AMC.

1. Open AMC (on local computer).
2. Right click the **Computer** node and select **Add/Remove Computers**. The **Computers** dialog will be displayed.





3. Type the name of the remote computer hosting Agresso Server software in the **Available Computers** field and click the **Add** button. This can be repeated if you want to add more than one computer.

Note: There is no functionality that allows you to browse for the remote computer.

The added computer names will be transferred to the **Added computers** list.

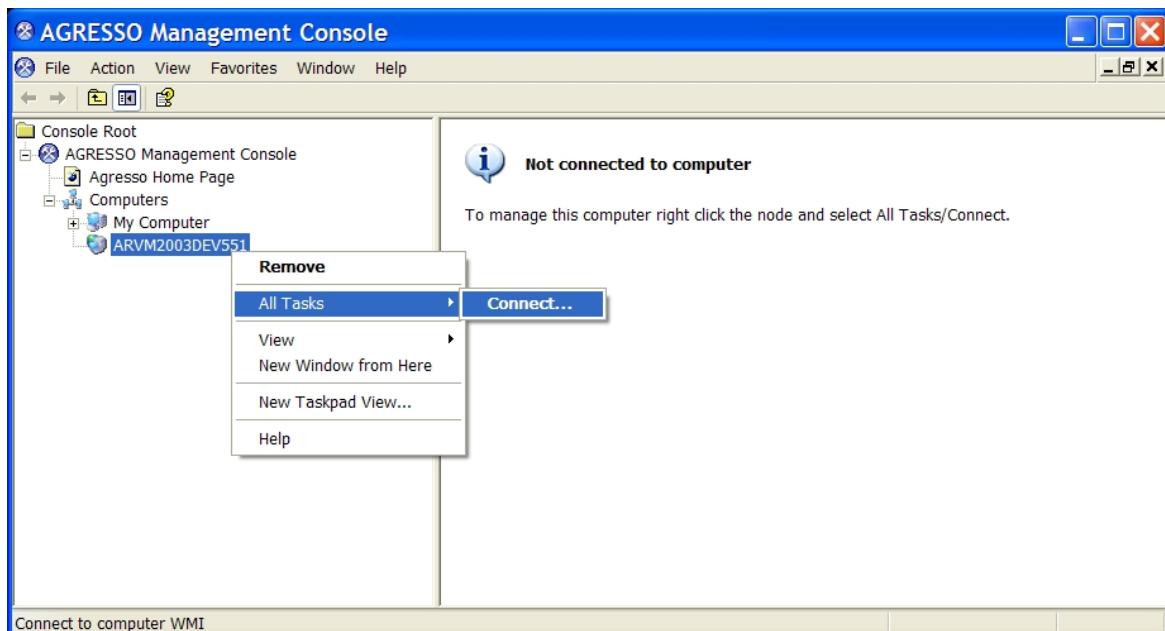
4. Click **OK** to add the new computer(s) to your local AMC.

Connect to the remote computer

Do as follows:

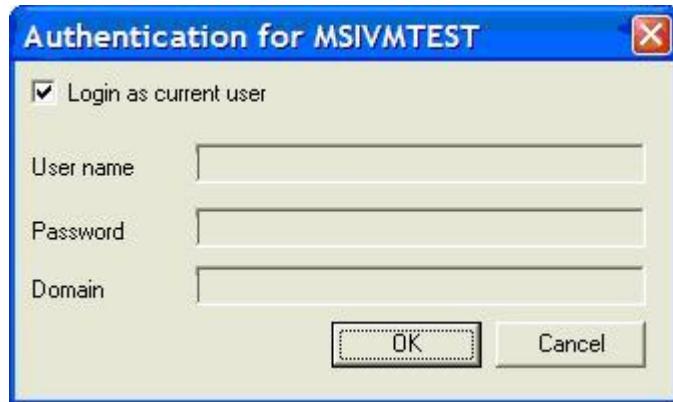
1. Right-click on the (newly added) computer node in the AMC node pane, and select **All Tasks/Connect**.

Step example: Connect to Remote Computer



As a result, the **Login** dialog will appear.

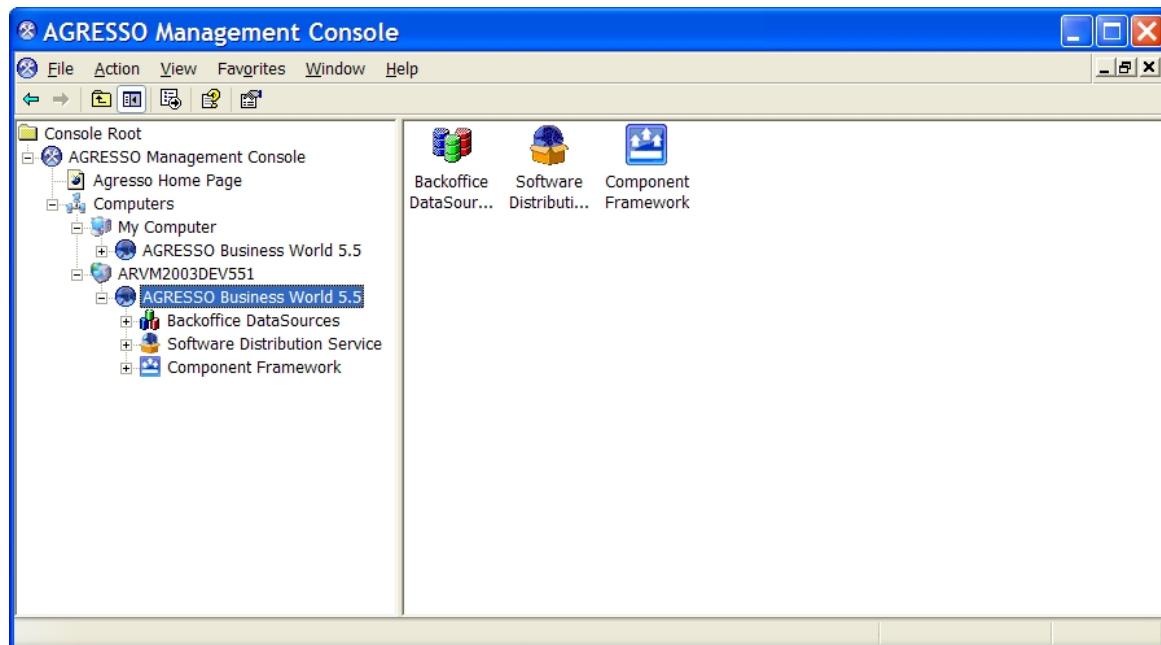
Example: Login Screen



2. Log in as a user with sufficient security clearing.

Possible errors: The **Agresso Management Console** will now try to connect to the remote computer's WMI service. If the server is turned off, or unavailable, an error message will display. The same will happen if you log in as a user with insufficient rights.

Success: Once connected to the remote WMI, the Agresso Installation window will appear:



You can now administer the Agresso installation from your local computer.

CONFIGURATION

Please read the general introduction to Agresso's Logging System

Before you start with this topic, make sure you are familiar with the Agresso Logging System, as described in the document [Logging in Agresso](#) in the Appendix.

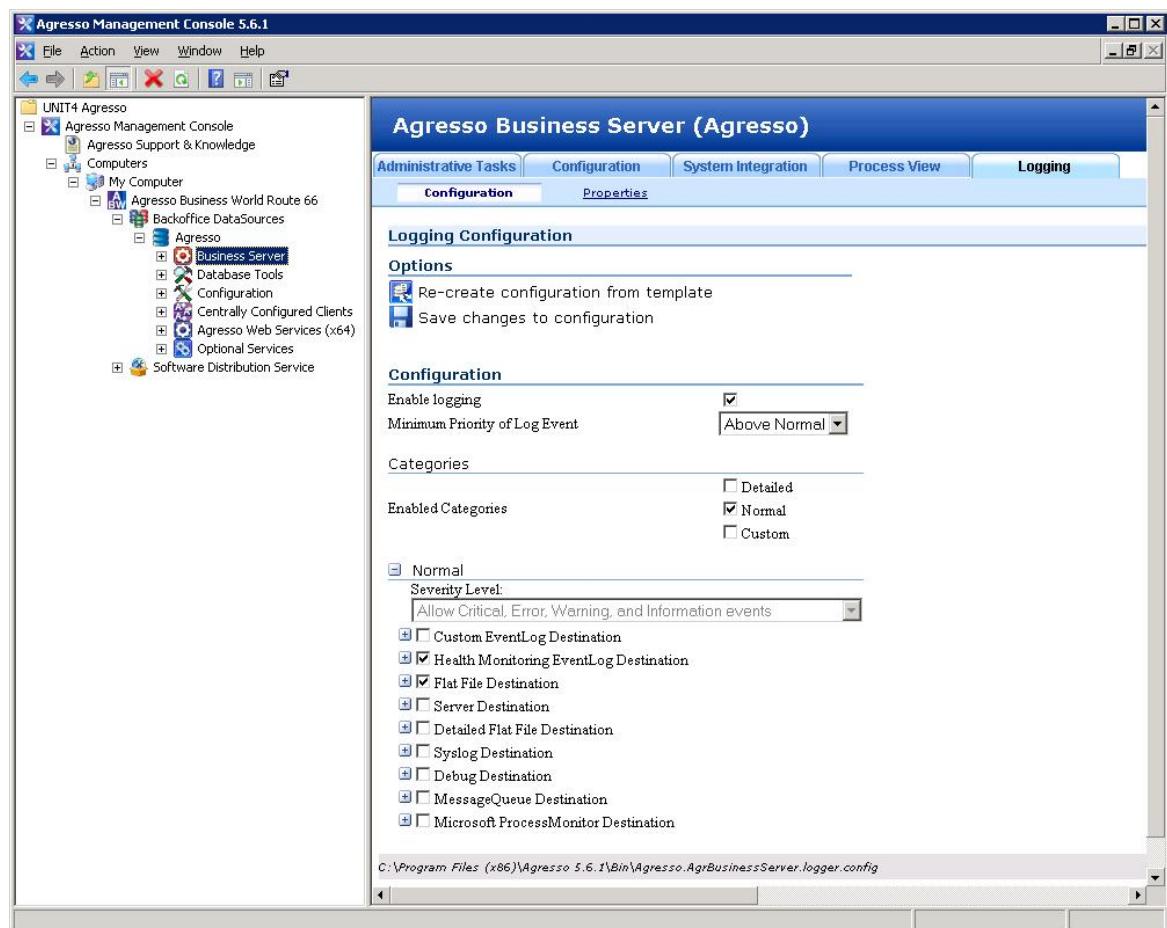
The logging configuration page

You use AMC to set up general (or global) logging options (see previous topic, [General Logging Setup](#)), as well as detailed configuration for certain processes and programs, available as nodes in the management console.

First, when you have decided to enable logging, you must enable logging for the business server. AMC will provide you with templates for default logging setup, but this can be modified. Next, you can enable and modify logging for all server queues of interest, Web services, as well as a few additional programs running at the server.

For all processes, services and programs you add to your Agresso installation, you will use the same page (or screen) for logging configuration, available behind the **Logging** tab associated with the AMC node:

Logging Configuration page (example)



Note: The [Properties](#) link shown in the example above, is only available for the **Business Server** node!

For more details and examples, see [Logging scenarios](#).

DATABASE LOG MESSAGES

Description

The **Database Log Messages** node displays information found in the log table `acreperror`. It also allows you to easily filter out relevant information.

You find **Database Log Messages** under the **Business Server** node and under each server queue node in AMC. When the node is accessed from under a server queue, you will by default only see errors for the current server queue. When the node is accessed from under the business server node, you will see all the latest log entries.

Log tables

The log tables are described below:

Table name	Description
acreperror	Contains errors that occurred when running reports. Not all errors are written to this table, so you should view the relevant log file for detailed error information.

Note: This table may be cleared at regular intervals, depending on the clean-up settings.

Options

Under the **All tasks** context menu, you have several options to filter out recent log entries. In the **Properties** dialog of the node you can for instance add custom filters to filter out errors for specific reports.

By default you will only see the latest log entries. The first row in the database log viewer will be the latest log entry - set by the **last_update** column.

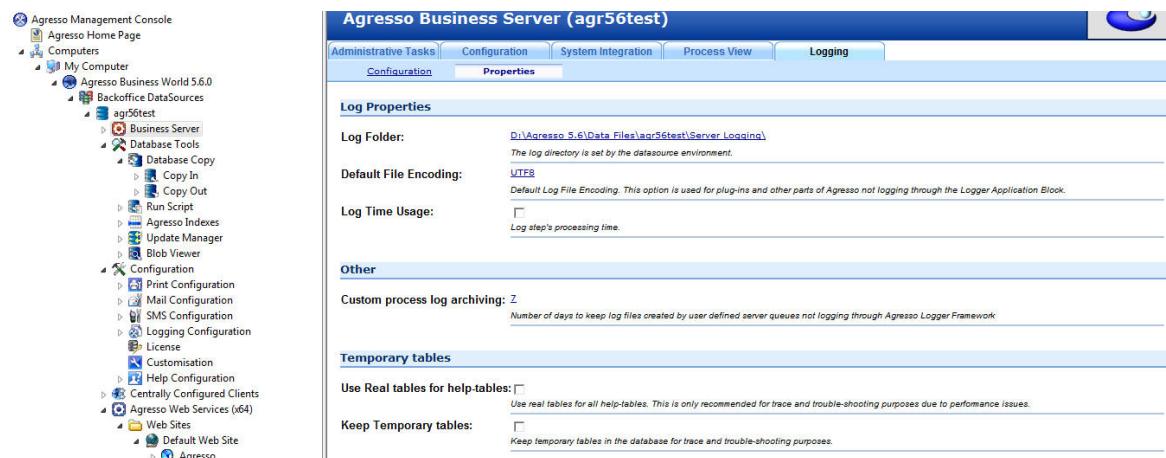
Logging

GENERAL LOGGING SETUP

Properties: Basic logging options

You use the **Properties** link on the **Logging** tab (**AMC**, **Business Server** node) to enter values for a few, basic logging parameters.

Setting Logging Options (example)



Log properties

Field name	Description
Log Folder	The default folder for logging. The value is stored in the environment variable AGRESSO_LOG. When logging is initialized, the AGRESSO_LOG directory will be used by default for log files created by the Business Server. If the AGRESSO_LOG environment variable is modified, the logging configuration will need to be updated to use this directory; this can be achieved by re-initializing logging. The "Server destination" logging destination used by the report queue, will always create log files in the directory defined by AGRESSO_LOG.
Default File Encoding	The value is stored in the environment variable AGR_LOG_FILE_ENCODING. Default value is UTF-8 .
Log Time Usage	The log file will show the time each task (query) uses. This might be useful during optimization.

Other

Only relevant if you create your own server queues.

Temporary Tables

Field name	Description
Use Real tables for help tables	Set the server processes to use real database temporary tables. Real temporary tables are not logged by the database and are therefore faster.

	<p> The service will always be logging to the file Agresso Server 5.6-<datasource name>.log. This log is meant for extended history information as well as for troubleshooting purposes.</p>
Keep Temporary tables	<p>A server process creates one or more temporary tables (working tables) during its lifetime. The tables are by default deleted during the process' clean-up phase. Mark this check box if you want to keep the working tables. Use this setting only on special occasions, for instance on request from your support provider.</p> <p> This setting should only be used for troubleshooting purposes.</p>

CONFIGURATION

Please read the general introduction to Agresso's Logging System

Before you start with this topic, make sure you are familiar with the Agresso Logging System, as described in the document [Logging in Agresso](#) in the Appendix.

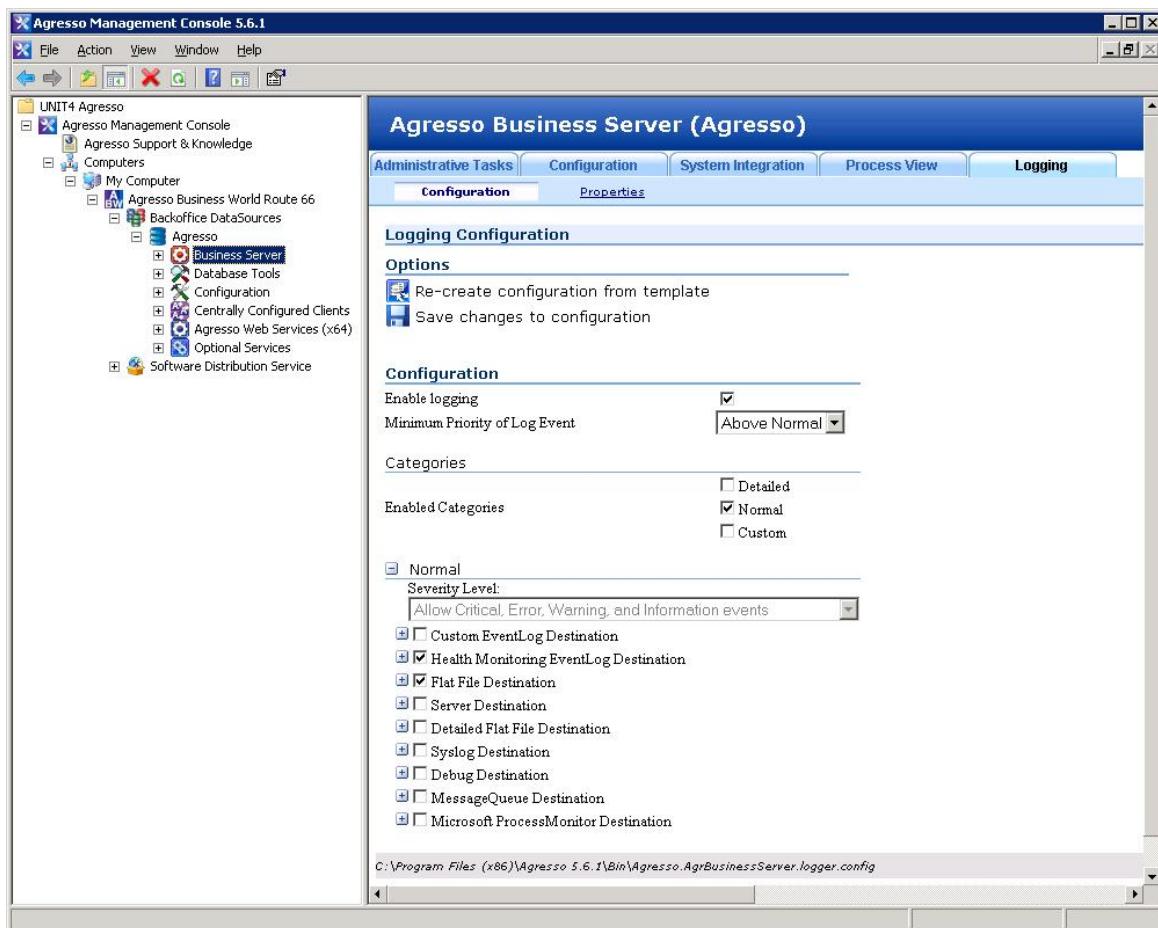
The logging configuration page

You use AMC to set up general (or global) logging options (see previous topic, [General Logging Setup](#)), as well as detailed configuration for certain processes and programs, available as nodes in the management console.

First, when you have decided to enable logging, you must enable logging for the business server. AMC will provide you with templates for default logging setup, but this can be modified. Next, you can enable and modify logging for all server queues of interest, Web services, as well as a few additional programs running at the server.

For all processes, services and programs you add to your Agresso installation, you will use the same page (or screen) for logging configuration, available behind the **Logging** tab associated with the AMC node:

 [Logging Configuration page \(example\)](#)



Note: The Properties link shown in the example above, is only available for the **Business Server** node!.

For more details and examples, see [Logging scenarios](#).

LOGGING SCENARIOS

In this topic, we will go through some typical scenarios and show how you use the Logging configuration options to set up the logging system for each scenario. First, we describe the default logging setup.

See [Logging in Agresso](#) for a description of the general options.

Default logging

Example

During installation, Agresso Business World will automatically be set up with "normal" logging. We show the default setup for the Agresso Business Server as an example:

Configuration

Enable logging

Minimum Priority of Log Event Above Normal ▾

Categories

Detailed

Enabled Categories Normal
 Custom

Normal

Severity Level:

Allow Critical, Error, Warning, and Information events ▾

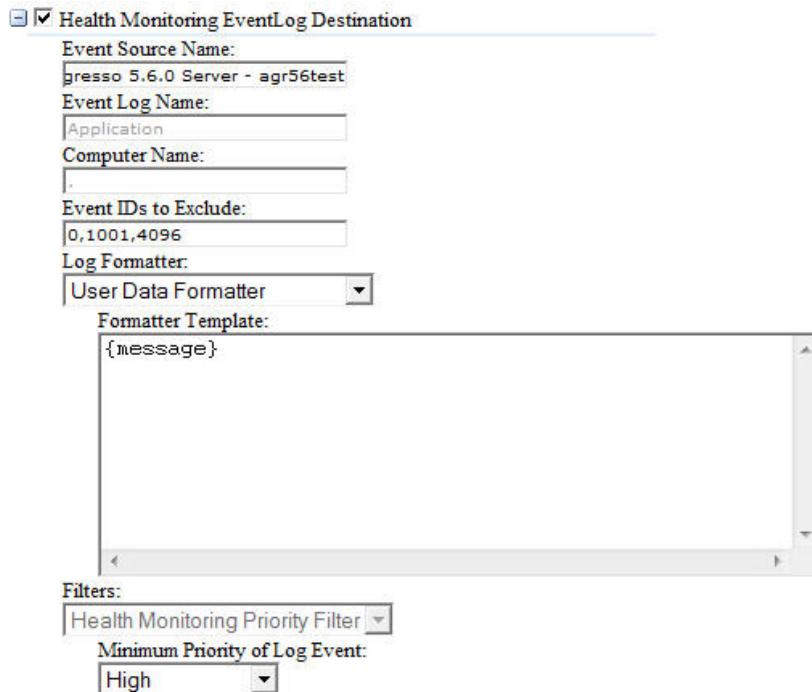
- + Custom EventLog Destination
- + Health Monitoring EventLog Destination
- + Flat File Destination
- + Server Destination
- + Detailed Flat File Destination
- + Syslog Destination
- + Debug Destination
- + MessageQueue Destination
- + Microsoft ProcessMonitor Destination

With this setup, all "loggable" events with a priority set to **Above Normal** or higher, will be taken care of by the **Normal** log category, and sent to two destinations:

- The **Health Monitoring EventLog**, and
- a **Flat file**.

EventLog destination

When we open destination details for the EventLog, we find the following details:



Agresso events

All the specific Agresso events are described in the Web application [Agresso Event IDs](#) (also available in the Appendix).

The **Event Source Name** is by default given the name of the current data source. This is a text you can modify. It will be used to mark the Agresso Events in the log.

You can also modify the content in **Event IDs to Exclude**, and change the formatter and formatter template.

What is logged, and to where?

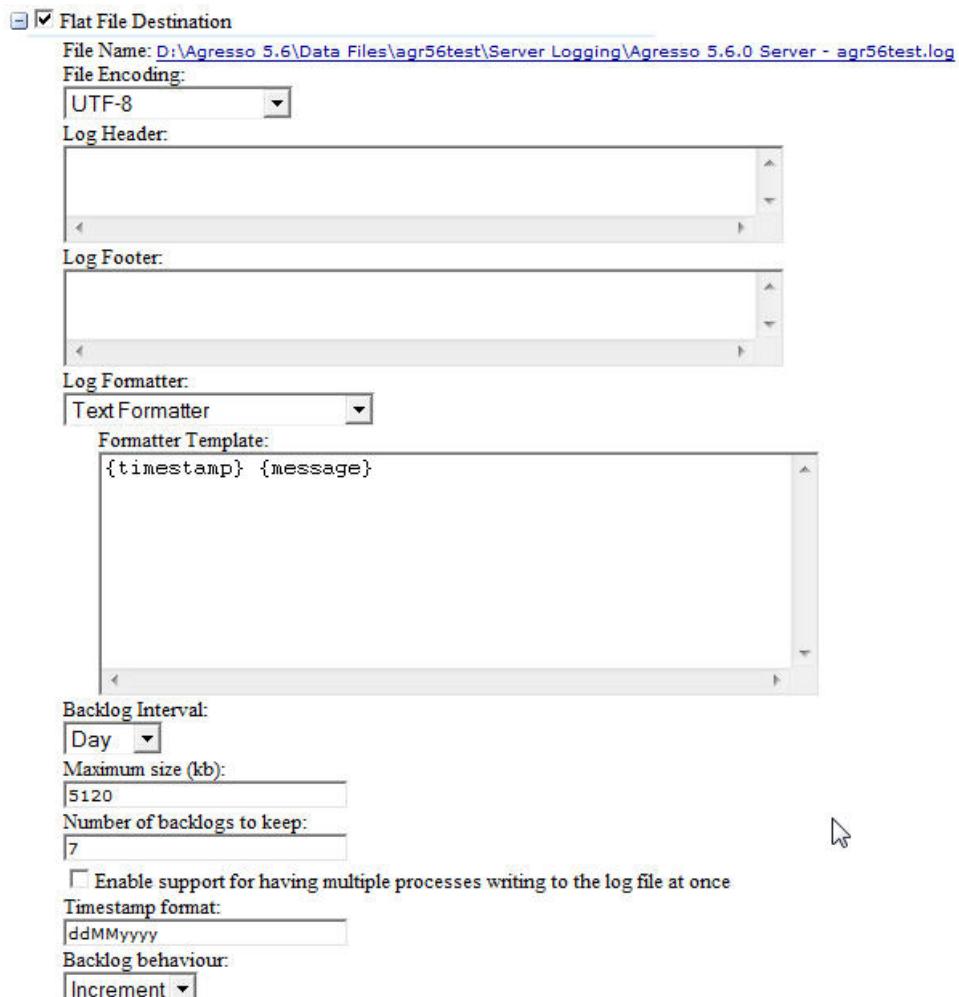
Agresso Events are logged as formatted entries in Windows Event Log.

Monitoring systems

You can set up any suitable third party monitoring system to monitor the Agresso application, based on log entries marked by the **Event Source Name**.

Flat File destination

You have more options when you look at the details for the Flat File destination:



We will explain the last fields (not covered in [Logging in Agresso](#)).

Field name	Description
Backlog Interval	Means how long the same log file shall be used, before the content is written to a backup file (or overwritten - see <i>Backlog behaviour</i> below), and the log-file is emptied. You can select None (no backup), Hour , Day , Week , Month , or Year .
Maximum size (kb)	Maximum size before the file content is written to backlog file (or overwritten).
Number of backlogs to keep	Number of backup files to keep before the first one is overwritten (new round).
Enable support for ...etc	If selected, several processes can write to the same file. Not recommended unless you know that the number of loggable events are few.
Timestamp format	Date format in log file.
Backlog behaviour	You can select Increment or Overwrite . Increment is the only logical value if you want to keep a backlog.

What is logged, and to where?

In the default configuration, log entries consist of a timestamp and the log message, and they are written to the named log file (File Name:).

Logging and troubleshooting

When you detect repeated, unexpected behaviour, you will probably need as much information you can get, in order to troubleshoot properly. The settings below shows a normal setup for troubleshooting:

Configuration

Enable logging	<input checked="" type="checkbox"/>
Minimum Priority of Log Event	Low

Categories

Enabled Categories	<input checked="" type="checkbox"/> Detailed <input type="checkbox"/> Normal <input type="checkbox"/> Custom
--------------------	--

Detailed

Severity Level: Allow Critical, Error, Warning, Information, and Verbose events

- + Custom EventLog Destination
- + Health Monitoring EventLog Destination
- + Flat File Destination
- + Server Destination
- + Detailed Flat File Destination
- + Syslog Destination
- + Debug Destination
- + MessageQueue Destination
- + Microsoft ProcessMonitor Destination

With this setup, all events are logged and written to a file. Note the Severity level, which now includes Information and Verbose events.

Database Tools

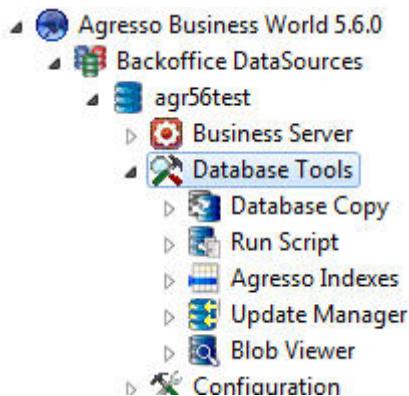
DATABASE TOOLS OVERVIEW

Tools

Agresso Management Console offers a set of tools for maintaining the database. These are

- **Database Copy**, offers tools for copying data to and from an Agresso database.
- **Run Script** (Agresso SQL Scripts), allows you to run scripts in Agresso SQL (ASQL) directly from the Management Console.
- **Agresso Indexes**, allows you to regenerate Agresso indexes.
- **Update Manager**, allows you to run specific update scripts enclosed in dlls.
- **Blob Viewer**, allows you to browse, view and delete blobs in the Agresso database.

Where you find them



[Database Copy](#)

Reference

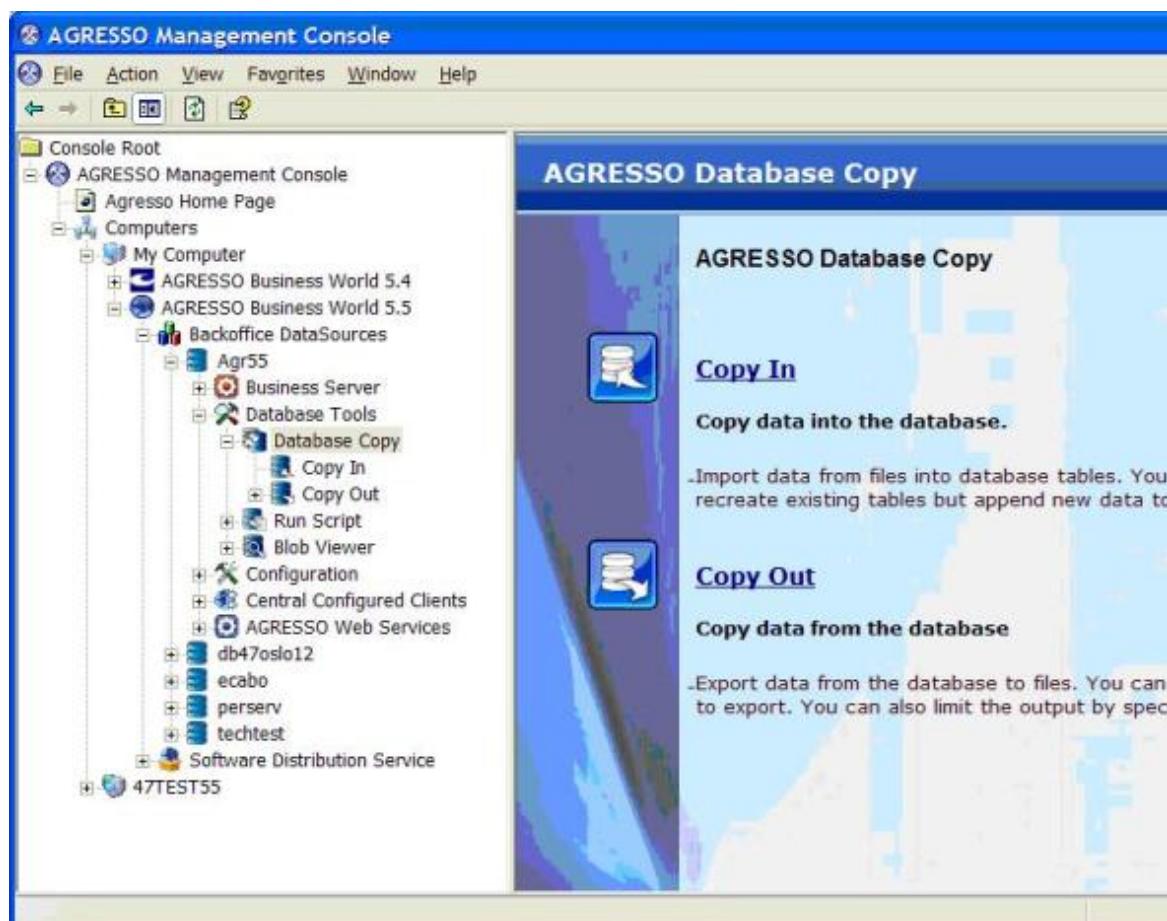
For all details about Database Copy, please refer to the documentation starting with [Agresso Copy](#) in the Appendix.

Purpose

Agresso Copy is used to load tables into a new Agresso database (Copy In) and to copy tables from one database to another (Copy Out). The tools are based on the same functionality as the CopyMS and CopyOra command line based tools. AMC adds a user-friendly layer on top of the basic tools to make them easier to use.

Note: Some of the more advanced options found in the basic tools might not be available from within AMC.

Database Copy



Establish database connection

You can establish a database connection without being validated as an Agresso user. This is to support a situation where you are setting up an Agresso database from scratch or repairing databases where the Agresso user or security tables are inconsistent and Agresso validation is not available. As with the command line based tools you need the database user and password to establish the connection. This is done automatically for you if you are already logged in at data source level.

- **Log in with validation:** To log in as an Agresso user, select the Data Source node and enter your Agresso credentials.
- **Log in without validation:** To log in without being validated as an Agresso user, go to the **Database Copy** node, right-click the **Copy In** or **Copy Out** node and select **All Tasks | Login**.

Run Script

Reference

For all details about the Agresso SQL Scripts (ASQL), please refer to the documentation starting with [Agresso SQL Program Overview](#) in the Appendix.

Purpose

The **Run Script** database tool is used when you need to execute SQL statements written in either Agresso SQL syntax or native database syntax. The statements are stored in one or more ASCII file with **.asq** file extension.

Establish database connection

As for Database Copy, you have two log in options:

- **Log in with validation:** To log in as an Agresso user, select the Data Source node and enter your Agresso credentials.
- **Log in without validation:** To log in without being validated as an Agresso user, right click the **Run Script** node and select **All Tasks | Login**.

Update Manager

The Update Manager is a general tool for database updates of Agresso system tables between service packs. The updates are packed as version controlled dlls, and will either come on the installation DVD (for example: Localisation updates) or be made available from our web site.

The use of Update Manager is quite straightforward: First, you browse for the dll and add it to the **Update Manager** node. Next, you run the update. For details, see [Update Manager Overview](#)

Blob Viewer

The Agresso Blob Viewer allows you to browse for, view and delete blobs found in your Agresso database.



AGRESSO Management Console

File Action View Favorites Window Help

Database Tools

- Database Copy
- Run Script
- Blob Viewer**
 - aagblob(0)
 - aagcerts(0)
 - aagdistrpar(0)
 - aagtmcchart(0)
 - acramendhist(0)
 - acramendlog(0)
 - acramendloghd(0)
 - acrdistrpar(0)
 - acrmessagequeue(0)
 - acrmessagequeuehead(0)
 - acrprintblob(6)
 - acrxmlimport(0)
 - actblob(29)
 - actcrasm(0)

description	blob_size	blob_id	File Type
Fieldhelp sales order	61.83 KB	79	Text
Fieldhelp location	20.81 KB	78	Text
Fieldhelp warehouse	19.45 KB	77	Text
Fieldhelp unit	8.06 KB	76	Text
Work orders per project	15.25 KB	74	Text
Warehouse orders	14.38 KB	72	Text
Contract	38.11 KB	71	Text
Product	36.07 KB	70	Text
Requisition	28.75 KB	69	Text
Field help Projects and draft projects	26.24 KB	68	Text
User - time statistics	11.24 KB	66	Text
Process - time statistics	8.70 KB	65	Text
Agresso.cer	1.27 KB	64	application/x-pkcs12
ams.cer	1.36 KB	63	application/x-pkcs12
AgrVerisign.cer	1.03 KB	62	application/x-pkcs12
AgrFinans.cer	939 B	61	application/x-pkcs12

The Blob Viewer will display all the Agresso tables containing blobs. When you select a blob table node, AMC will list all rows found in the blob table in the right pane. Selecting one of these nodes enables you to execute, save or delete the blob you have selected.

Database Copy

COPY IN

Main functions

Copy In is used to copy data and indexes stored in files into the database.

The following functions are available:

- Start Copy
Starts the Agresso Database Copy Wizard. This option is only available when one or more files are selected.
- Re-run Copy
Starts the Agresso Database Copy Wizard. But the **Finish** button is enabled from the first page to start the copy operation using the same options used in your previous execution.
- Re-create views
Creates/re-creates views defined in the tables `asysviews/aagviews` without copying additional data. No selection is required to do this.
- Add From List File
Add a predefined selection of data files to the selection list. The function let you to browse for an .lst file containing a list of datafile names. All files listed in the .lst file will be populated in the selection view (right pane).
- Add File(s)
You are able to browse for one or more data files and add them to the selection list.
- Logout
Description: Terminates the database connection.

Copy to database

You typically start out adding a selection of data files to be copied into the database. You can add a predefined selection by selecting a list file (.lst) or by adding individual files. You can manipulate the selection by adding additional files or by deleting one or more selected items.

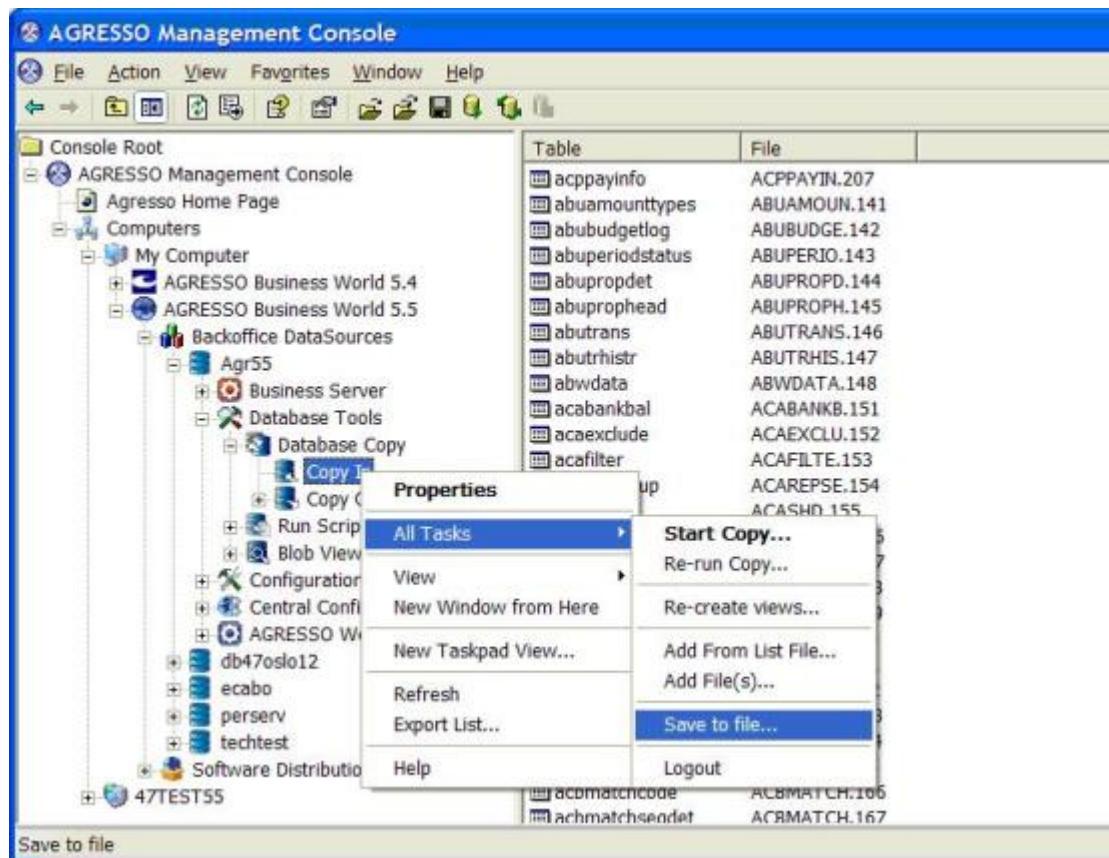
First, select the files (use Shift or Ctrl to select more than one file). Then press the Delete key or right-click the selection and click **Delete**.

To view the content of a file, double-click or right-click the table item and click **Open**. This will bring up Notepad or another editor to view the file. If you want to use an editor other than Notepad this can be configured by opening **Properties** on the **Copy In** node.

The selection can be saved by right-clicking the Copy In node and select "Save to File".

! *The last selection will automatically be saved to a file named "_cifiles.lst" when the copy operation is started.*

 Save selection



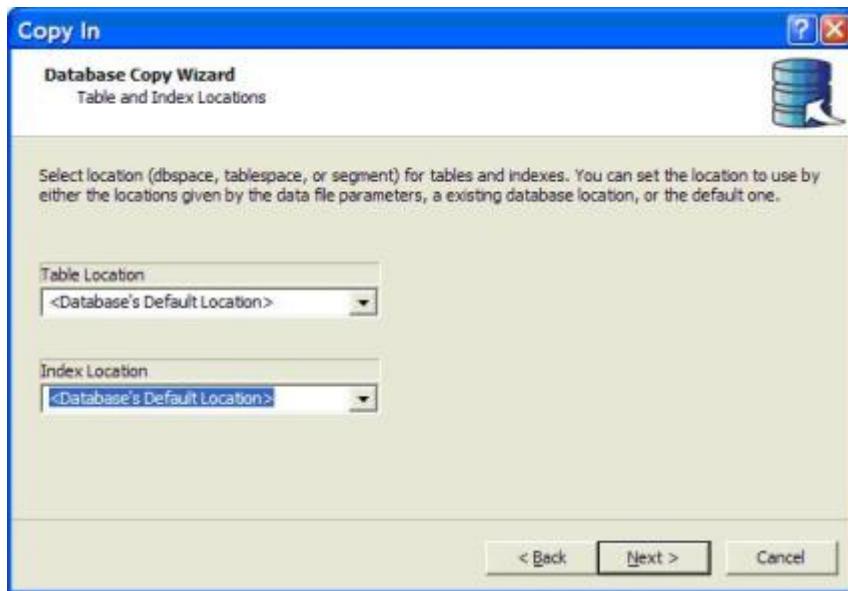
Copy In Procedure

1. To start the Copy In process, select All Tasks | Start Copy, or use "All Task | Re-run Copy" to repeat last copy out operation. This will bring up the Copy In wizard to control the Copy In process.

First, select Table and Index location. This is the dbspace, tablespace, or segment where the data is stored internally by the database. In addition to listing the specific table spaces there are two other options available:

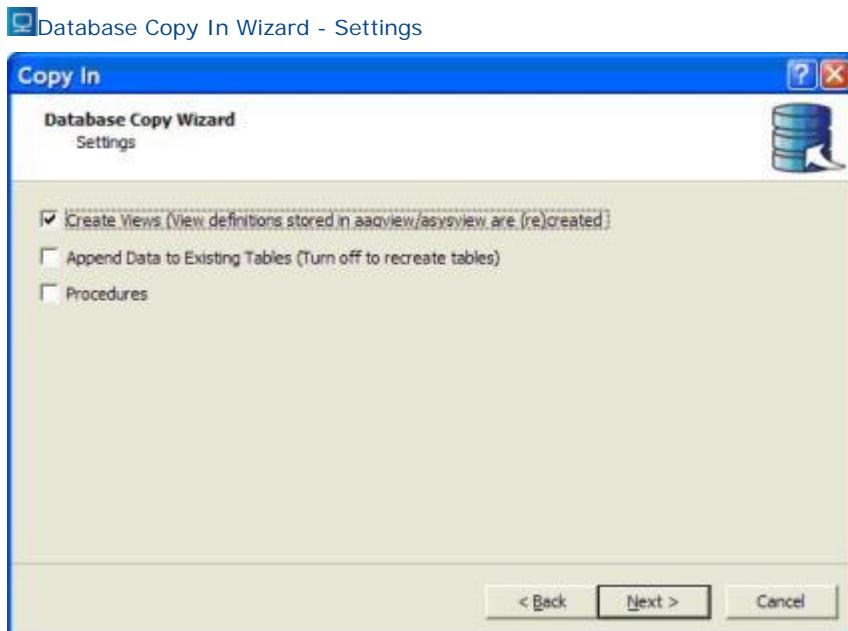
- <Database's Default Location>
Description: [Default] This will assure that all data is stored in the Database's default location.
- <Given by Datafile>
Description: This will attempt to use the same location name as the table or indexes was originally copied out from. This is given in [TABLE] section in the datafile:
[TABLE]
name=acabankbal
location=agrstatic

[Database Copy In Wizard - Table and Index Locations](#)



2. The next page is to set common options. Select/Deselect the following options:

- Create Views (-v)
Description: Create database views. View definitions are read from the table aagview.
- Procedures
Description: Procedures are created from definitions found in asysddl and aagddl.



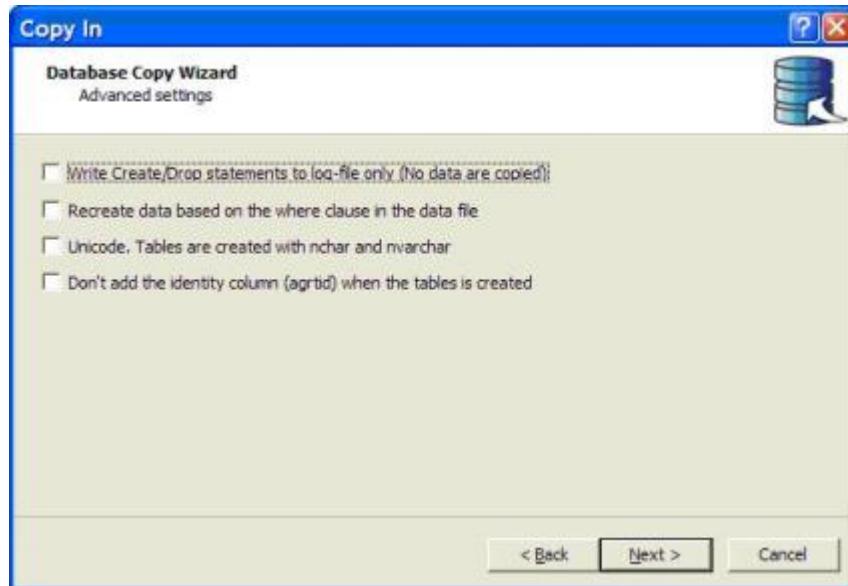
3. Next, select the advanced settings if applicable:

- Write Create/Drop statements to log file only
Description: Display Only. No tables are dropped or created and no data is read. You can use copy and paste from the output node, and later run the SQL commands from within interactive SQL.
- Recreate data based on the where clause
Description: Data in the table where the where statement is true is deleted. The data in the datafile is then added. The table is NOT recreated.
- Append Data to Existing Tables
Description: Instructs the copy program not to re-create tables before data is read into it.
- Unicode
Description: Tables are by default created using "char" and "varchar". The parameter "-u" results in tables created with types "nchar" and "nvchar".

- Do not add the identity column

Description: Used by MS SQL only. When this parameter is set, the identity column (agrtd) is not added when the table is created.

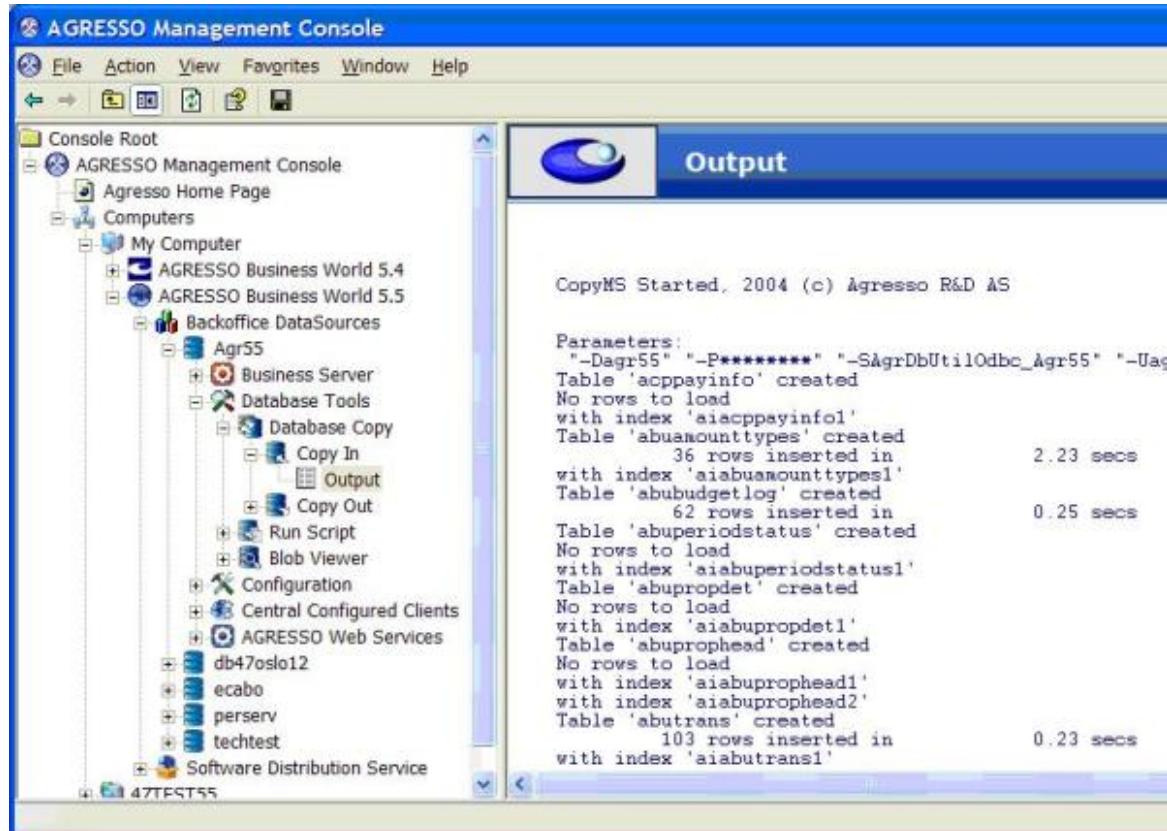
Database Copy In Wizard - Advanced Settings



When the wizard is completed, the Copy In process will start and output is written to an Output node during the operation. The processing will run in the background and AMC is not locked. You can continue with other tasks.

! You should not perform other copy operations during this time, as the some of the underlying functionality does not support this. A dialog will pop-up when the copy is finished.

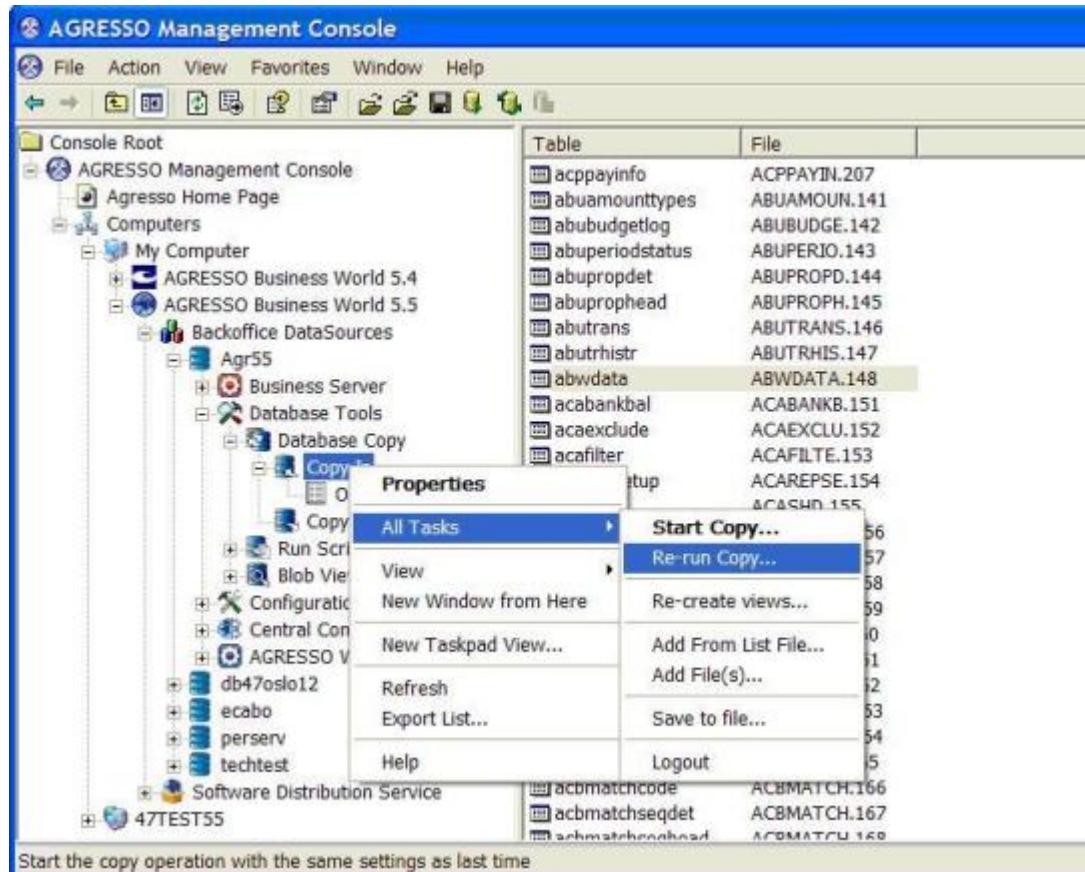
Copy In Output



Re-running Copy

After a copy operation is completed a new option is available on the Copy In node. Right-click and select "All Tasks\Re-run Copy" to quickly re-run the copy operations using the same options, or you can alter one or more options.

Re-run Copy



You can now start the copying process using the same options by clicking "Finish", or you can change some options by clicking "Next".

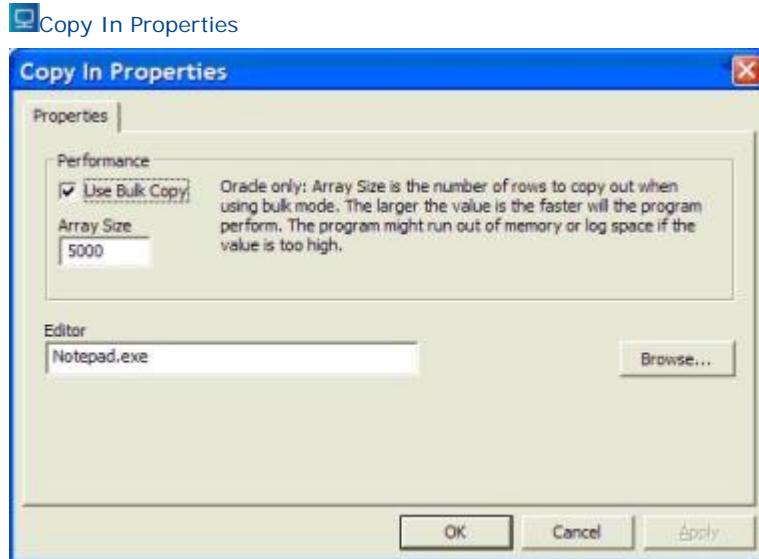
Data Copy Wizard



Copy In Properties

Properties other than the above are found as "Properties" on the Copy In node. They are:

- Use Bulk Copy [Oracle Only]
Description: Turn on/off use of Bulk Copy mode.
- Array Size
Description: This is the Array Size to copy out when using bulk mode.
- Editor
Description: Default editor to use when opening a data file from within AMC.



COPY OUT

Main functions

You will first make a selection of tables to be included in the copy operation. The selection can be defined in a .lst file, or you can browse the database for tables to include. You can then copy the selected table definitions and data to the file system.

The following main functions are available:

- Start Copy
Starts the Agresso Database Copy Wizard. This option is only available when one or more tables are selected.

Re-run Copy

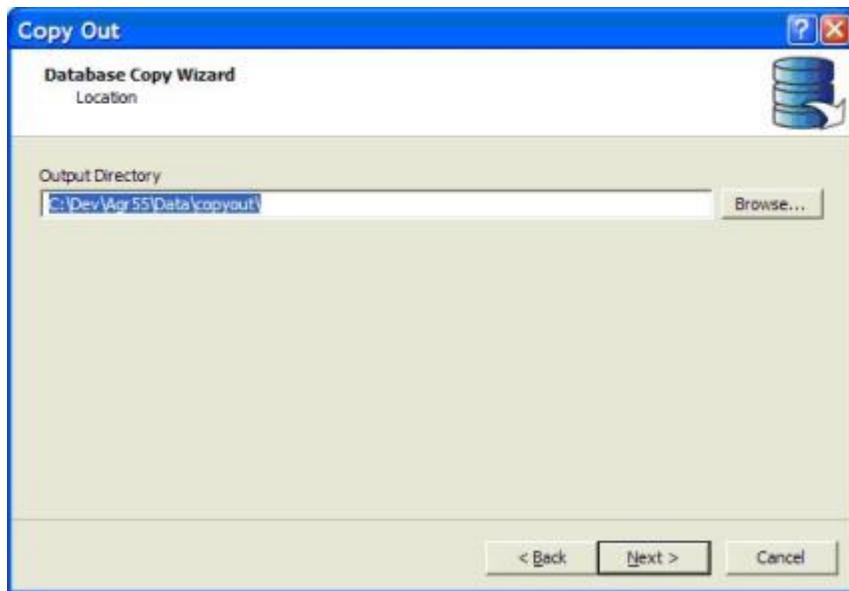
Starts the Agresso Database Copy Wizard. But the **Finish** button is enabled from the first page to start the copy operation using the same options used in your previous execution.

- Add From List File
Add a predefined selection of tables to the selection list. The function let you to browse for an .lst file containing a list of datafile names. All files listed in the .lst file will be populated in the selection view (right pane).
- Add From Database
Select tables directly from the database.
- Logout
Terminates the database connection.

Copy Out Procedure

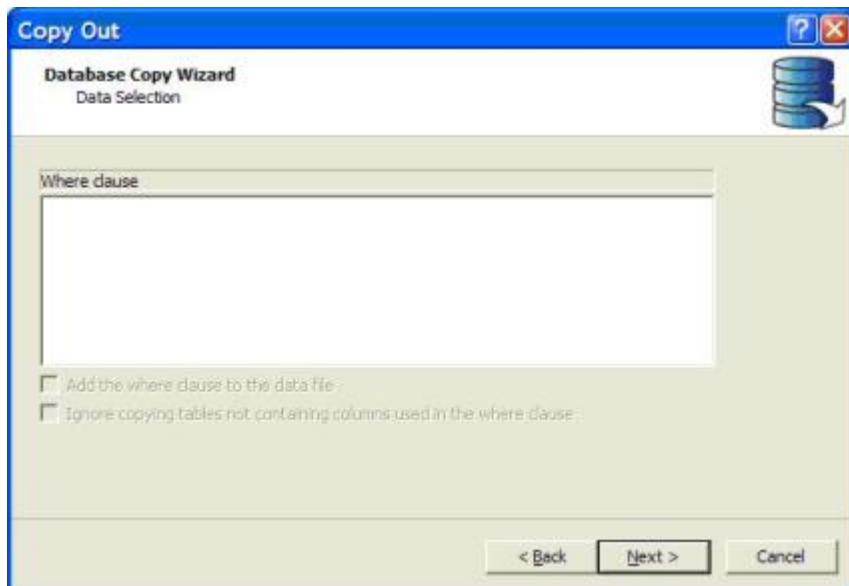
1. To start the Copy Out operation, right-click the **Copy Out** node in Agresso Management Console and select **All Tasks | Start Copy**. (To repeat the last copy operation, select **All Tasks | Re-run Copy**.)





2. Select the folder where the data files shall be written. The folder will be created if it does not exist.

[Database Copy Out Wizard - Data Selection](#)



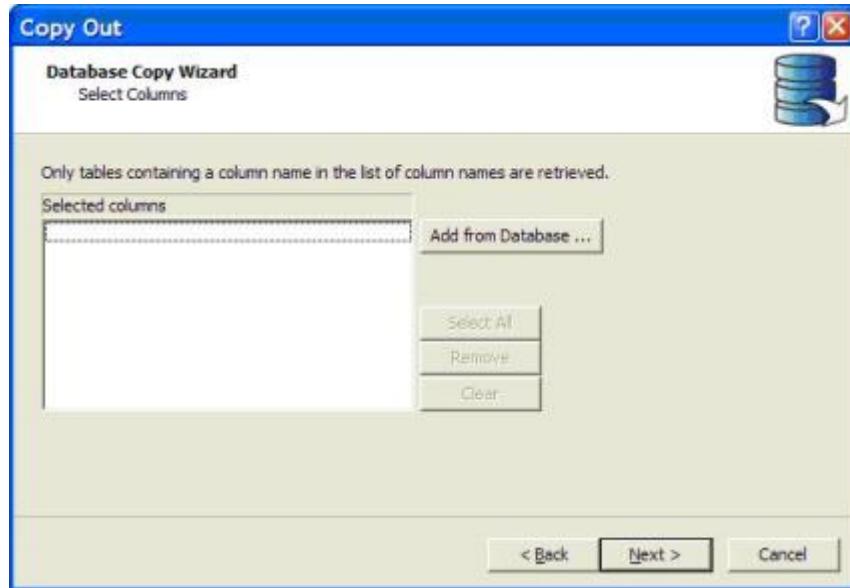
3. If you want to create an initial filter for table selections, you can enter a SQL where clause, for example
`apar_id like '1000%`'

Ignore copying Tables not containing columns used in the where clause

This can be used when for instance copying out tables only containing the column client and only rows with value EN.

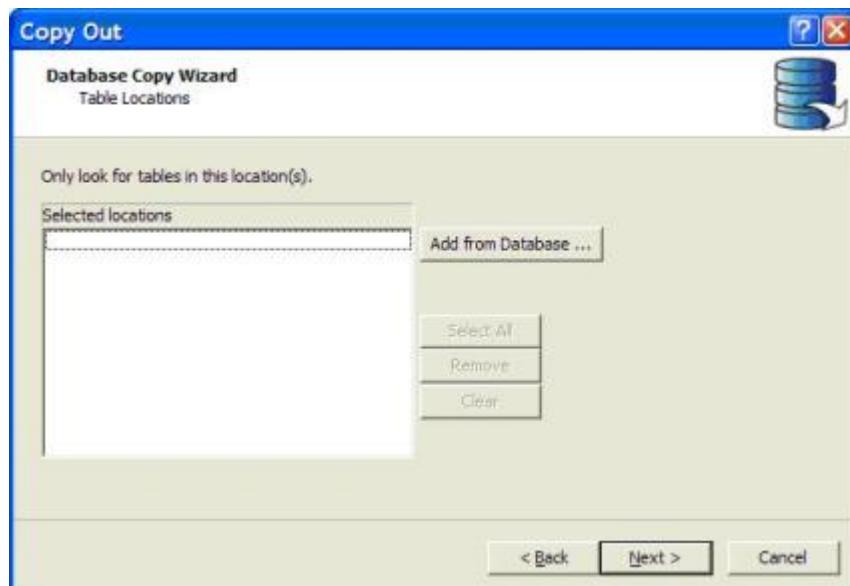
4. Next, you can limit the tables being copied out based on column name. Only tables containing a column name in the list of column names are retrieved. Leave empty for all tables.

[Database Copy Out Wizard - Select Columns](#)



5. Also, you can limit the tables being copied out based on table location. Only tables from the list of locations will be copied out. Leave empty for all tables.

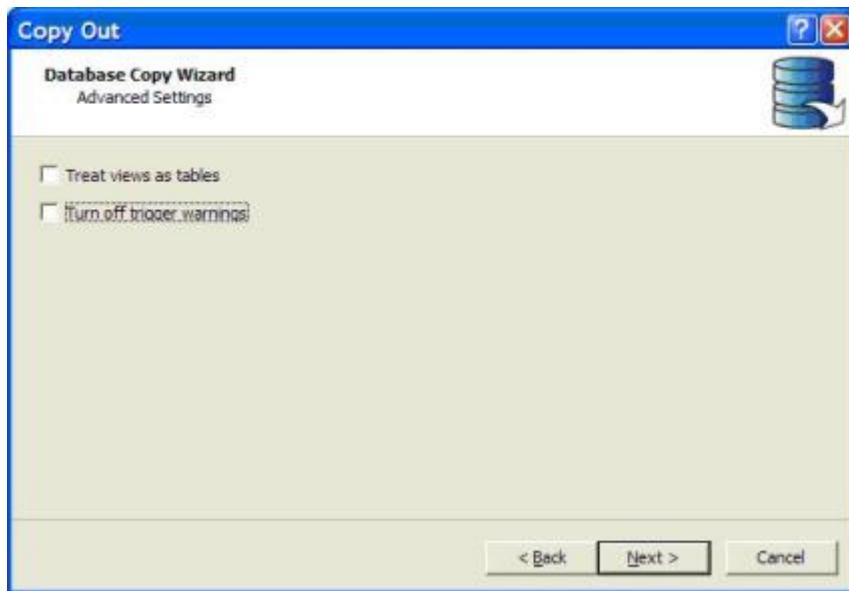
 [Database Copy Out Wizard - Table Locations](#)



6. Finally, the last page gives you the Advanced Settings:

- Treat views as tables
Description: Views are treated as tables, and copied out in the same way. Note, they can not be copied in as views again.
- Turn off trigger warnings
Description: If the table being copied contains a trigger, the definition is printed to the output window.

 [Database Copy Out Wizard - Advanced Settings](#)



RUN SCRIPT

Main functions

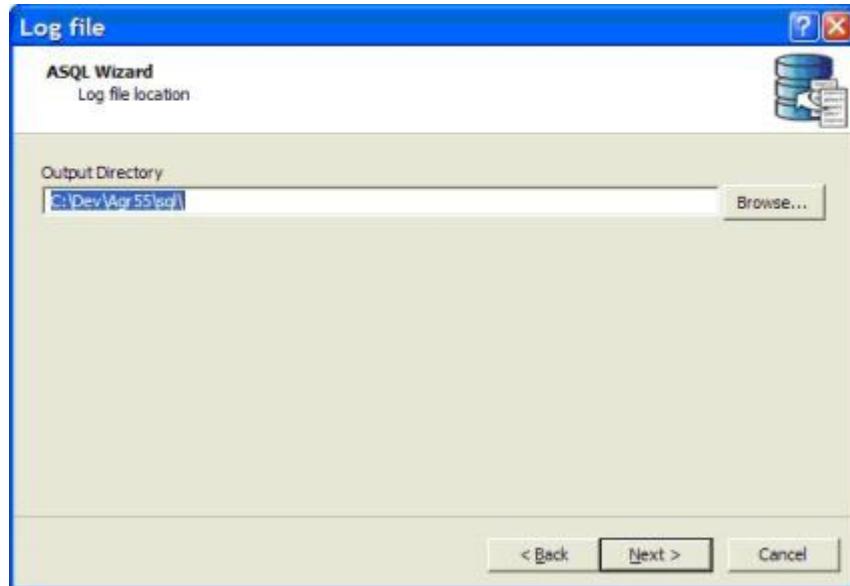
The following functions are available when right-clicking the **Run Script** node:

- **Start**
Starts the **Run ASQL Scripts Wizard**. This option is only available when one or more script-files are added to your selection.
- **Add From List File**
Add a predefined selection of scripts to the selection list. The function let you to browse for an .lst file containing a list of script-file names. All files listed in the .lst file will be populated in the selection view (right pane).
- **Add File(s)**
Browse for files to add to the selection.
- **Save To File**
Save your selections of script files (.asq) files to a List File (.lst)
- **Logout**
Terminates the database connection.

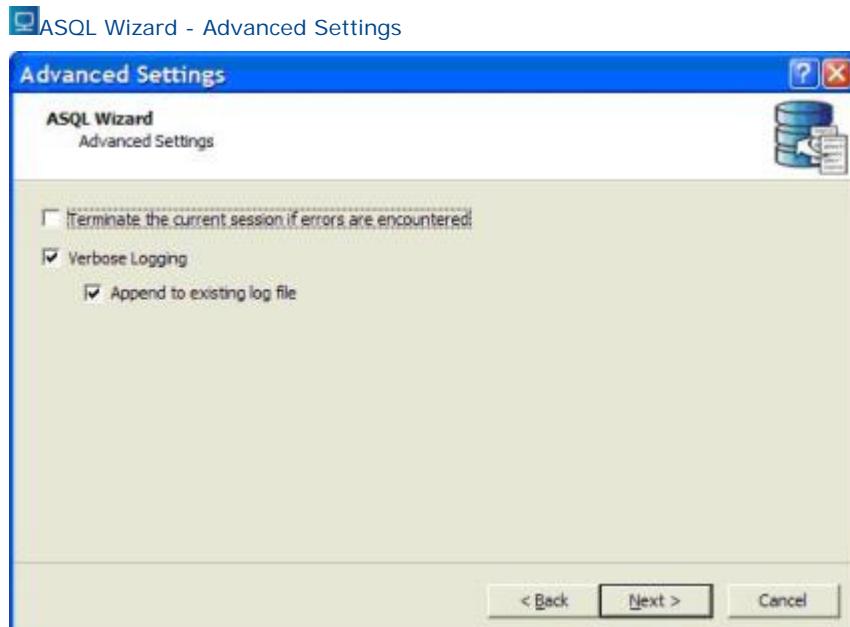
Run ASQL Scripts

1. To start an ASQL process, right-click the **Run Script** node in AMC and select **All Tasks |Add File(s)** to run the script.

 **ASQL Wizard - Log file location**



2. Select **Output Directory** where a detailed log file (*asql.log*) will be written during the run script operation. All commands and information are printed to this log file.



3. Do as follows:

- Select advanced settings (if required). The following options are available:
 - Terminate the current session if errors are encountered
This parameter is only used when running more than one asq file. It is used in conjunction with the "on error exit" command. The scope of the "on error exit" command is only the current file. Thus, if an error occurs after this command, *asql.exe* will skip the rest of the current file, but continue with the next one. However, if you want *asql.exe* to cancel all files when an error occurs, use this parameter.
 - Verbose Logging
Add more detailed logging to the *asql.log* file.
 - Append to existing log file
Do not empty the log file if it exists but append the log to the end of it.
- Click **Next** to start (and finish!).

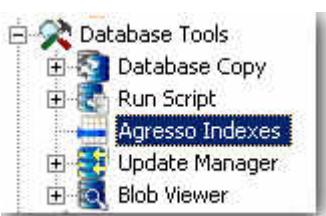
AGRESSO INDEXES

Description

The **Agresso Indexes** tools enables you to regenerate indexes defined in `asysindex` and `aagindex`. If a defined index is missing, the tool will re-create it for you. This tool will run the `agrindex.exe` application that also can be run from command line as described in the [appendix](#).

AMC location

You find Agresso Indexes under **Database Tools**:



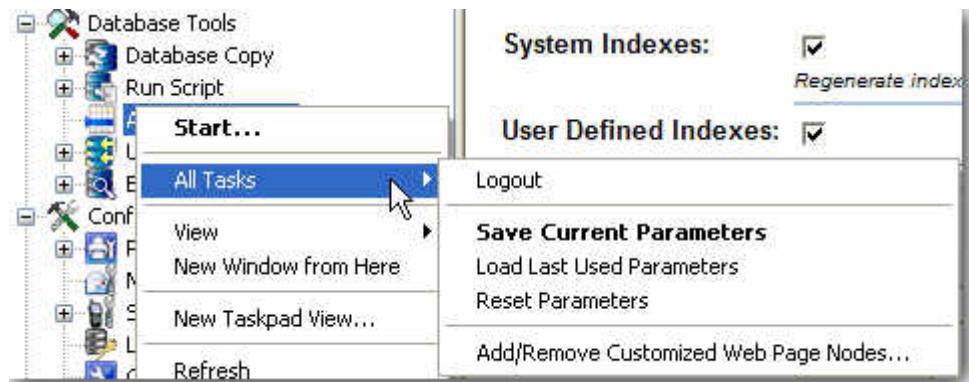
Execution parameters

When you select the **Agresso Indexes** node, you get access to the parameters available for the regeneration job.

Agresso Index Re-generator - example

Menu options

By right-clicking the node, the **All tasks** menu item displays the following options:



Update Manager

USING THE UPDATE MANAGER

Introduction

This topic covers the general use of the **Update Manager**, that is, how you add an update package.

Related topics

For information about update packages and steps, see [Update Manager details](#).

Adding an update package

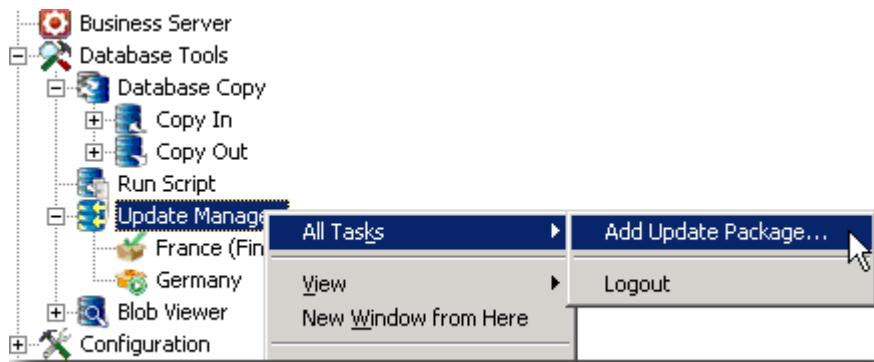
Update package files

The update manager works with update packages, one at a time, and an update package always comes as a .dll file (actually an `*.update.dll` file). The update packages can either come on an upgrade DVD, or they can be downloaded from a web site.

Procedure

Regardless of update category, the procedure for installing updates is the same. To add an update package (a dll), you do as follows:

1. Right click the **Upgrade Manager** node and select **All tasks | Add Update Package**.



Result: A standard Open dialog prompts you to browse for the dll with update scripts.

2. Locate and select the dll with updates.

Result: The package will be added to the **Update Manager** node:



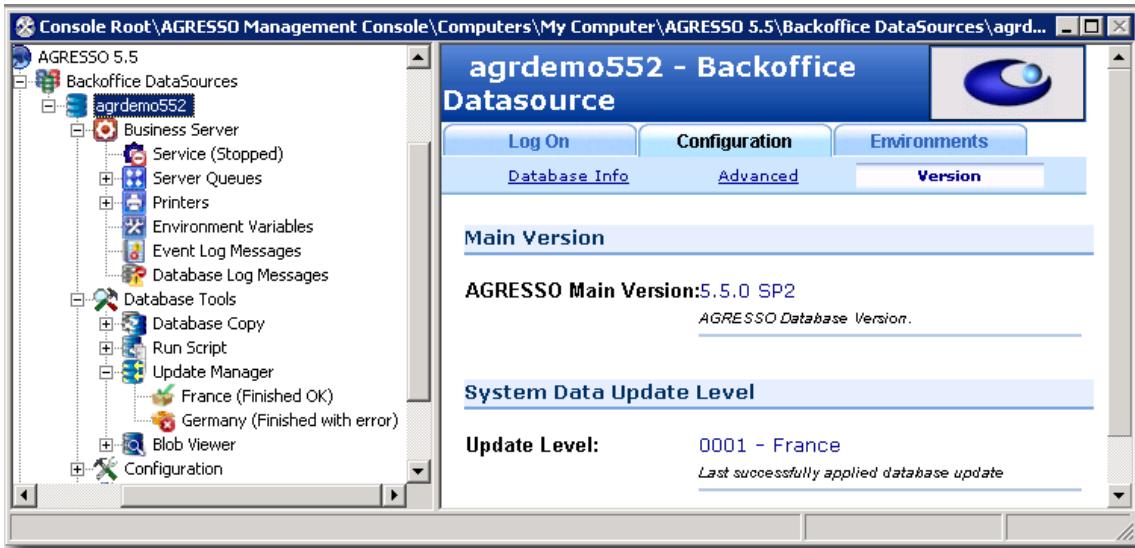
3. Right click the update you want to install, and select **All Tasks | Start ...**



Result: The **Update Manager** will execute the update scripts and give you a receipt when the work is done.

Database patches and version control

The **Data source** node (**Configuration** tab) in the **Agresso Management Console** contains detailed database version information:



The **Update level** property gives you the version and name of the last successfully applied update (patch).

UPDATE MANAGER DETAILS

Overview

This topic describes two important aspects of **Update Manager**:

- *The Update package*, consisting of update steps and properties.
- *The Database update storage* (database tables used by Update Manager)

The Update package

Package structure

An update *package* is the main object for the Update Manager, and contains one or more update *steps*. The diagram below shows the structure of three update packages, with various steps:

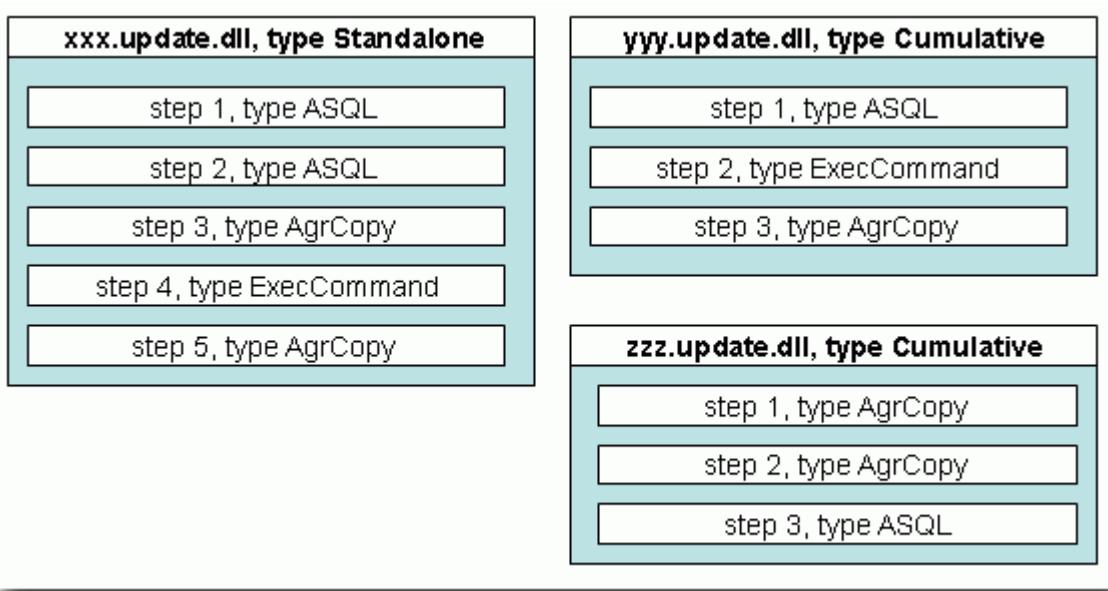


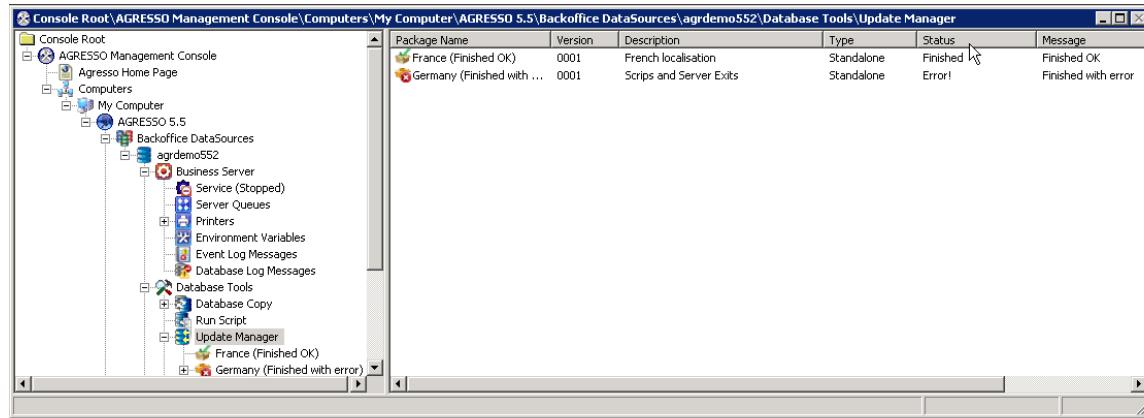
Diagram comments: The diagram illustrates that there are two types of packages, either Standalone or Cumulative. Cumulative package types will update the Update level property for the detailed database version, and cumulative packages must be added in the correct sequence. A cumulative package that requires an update to be run first, will have the status **On Hold**.

The diagram also illustrates that a step can be of any of three types:

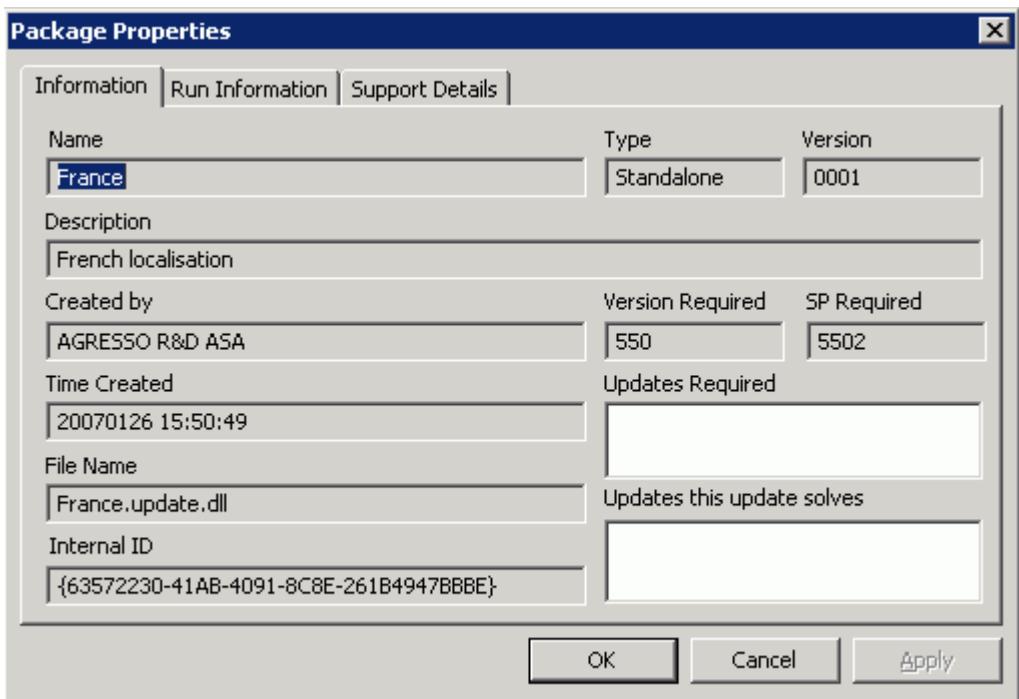
- ASQL - a script file
- AgrCopy - an Agresso Copy data file
- ExecCommand - a command file or an executable.

Package properties

When you open the **Update Manager**, the loaded updates (not necessarily executed yet) will be listed in the right AMC pane, like this:

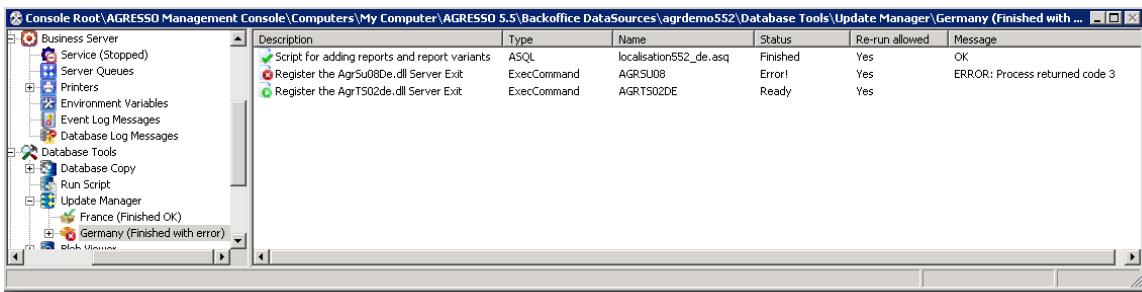


By right-clicking one of the package nodes, for example France, you can select **Properties** to get additional information about the selected package:



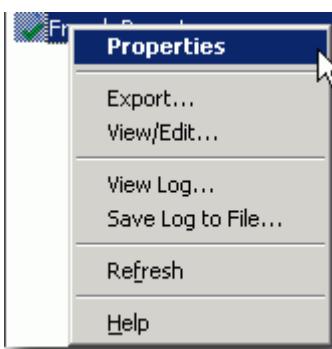
Step properties

General properties: By selecting an update package from the left AMC, the main properties for the various steps will be listed in the right pane - as shown below:



Context sensitive menu for steps

By right clicking a step, the following context sensitive menu appears:



Step properties: You can select **Properties** to get additional information about the selected step. The following properties are available for all step types:

Step property	Description
Edit of Step Allowed	You are allowed to open the Step for edit before run
Stop Update Package if	Flags if a Step is critical. An Update Package cannot finish if a critical step

an error occurs	fails. Contact Agresso Maintenance if this happens. Agresso Maintenance will send a new Upgrade Package to solve this issue.
Re-run of Step is allowed	The Step can be re-run without creating any harm to your data.

The following properties are available for steps of type AgrCopy only:

AgrCopy step property	Description
Re-create views	The step will re-create views based on definitions in the aagview when turned on.
Re-create procedures and triggers	The step will re-create procedures and triggers when turned on.
Use Where-clause in data file	The step uses a Where-clause in the AGRESSO Copy file to target update of special rows.
Append data only	AGRESSO Copy will only append data to the table and will not delete any existing rows.
Additional Parameters	Additional parameters used to control the AGRESSO Copy process.

Step actions: In addition to properties, the menu offers the following options:

- **Export** – Export the ASQL or **Agresso Copy** file to a local file
- **View/Edit** – View or Edit the step. You can modify the step, if the step allows changes.
- **View Log** – Opens the log file in the associated viewer
- **Save Log to File** – Export the log file to the file system.

Database update storage

Description

The **Update Manager** stores all update packages as blobs and keeps history and logs for all applied database updates.

The use of blobs assures that no information or files are stored locally, at the computer running AMC. Once an update package is added to the **Update Manager**, it is available from any computer running AMC. This also applies to the log files.

Tables in use

Update package information: The table **acupgradehead** holds header information about all database updates added along with status information.

Update step information: The table **acupgradedetail** holds detailed information about each step in the update package.

Log files and step modifications: The log files, as well as information about modified steps, are stored as blobs in the table **acupgradeblob**.

These tables may be collectively referred to as the *Database update storage*.

The Blob Viewer

BLOB VIEWER

View a blob

To view a blob, double-click on a blob node, or right-click on the node and select **View blob** from the menu.

If the file type of the blob you are trying to view is known, the blob viewer will try to execute the blob with the application associated with the file type extension. If the blob viewer does not recognize the blob you are trying to execute, or the execution of the file fails, you will be able to select a (new) view application.

All blobs that you try to execute will first be saved to the current log in user's temporary folder, and the Blob Viewer will then try to execute the file from that location.

Save a Blob

Using the **Save to file** option in the context menu, will enable you to save the blob to file at a custom location.

Delete a Blob

Delete blob

To delete a blob from a table, select the blob node you would like to delete, and press the **Delete** key. Alternatively, you can right-click on the node and select **Delete**.

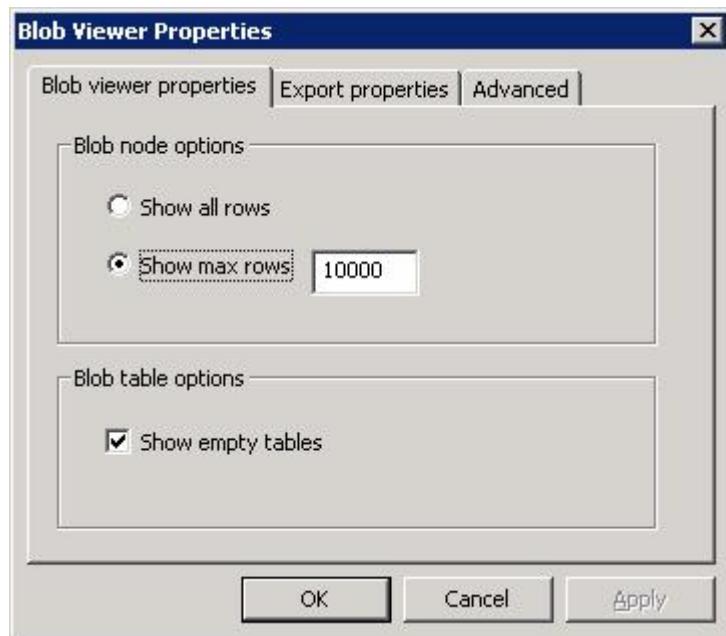
Options

Properties

You can open the **Properties** dialog for a blob viewer by right-clicking the blob viewer node, and select the **Properties** menu item.

The 5.5.2 release introduced new properties for Export, and now allows you to select where you want to get the table's metadata from - either the DBMS or the Agresso dictionary.

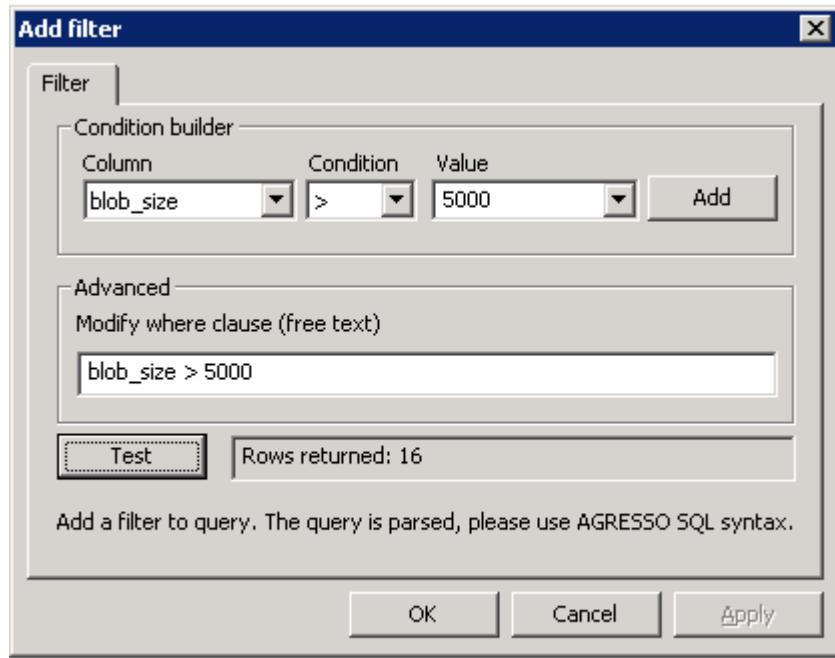
Example: Blob Viewer Properties dialog



Blob table filter

The **Add Filter** dialog allows you to add filters to the query used by the Blob Viewer to retrieve blobs from the database tables.

Example: Add Filter dialog



BLOB MANAGEMENT

Task overview

The **AGRESSO Management Console** gives you the following options:

- Insert new blobs
- Update existing blobs

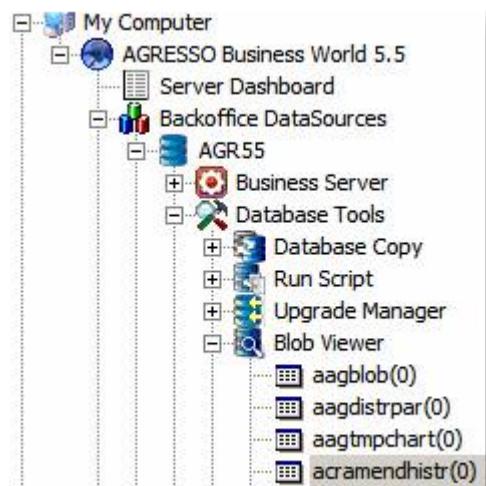
These tasks are described below.

Note: Any modification of system tables may cause problems for future upgrades. Be careful!

Insert blobs

To insert a blob, you do as follows:

1. Locate the **Blob Viewer** in **AGRESSO Management Console**.



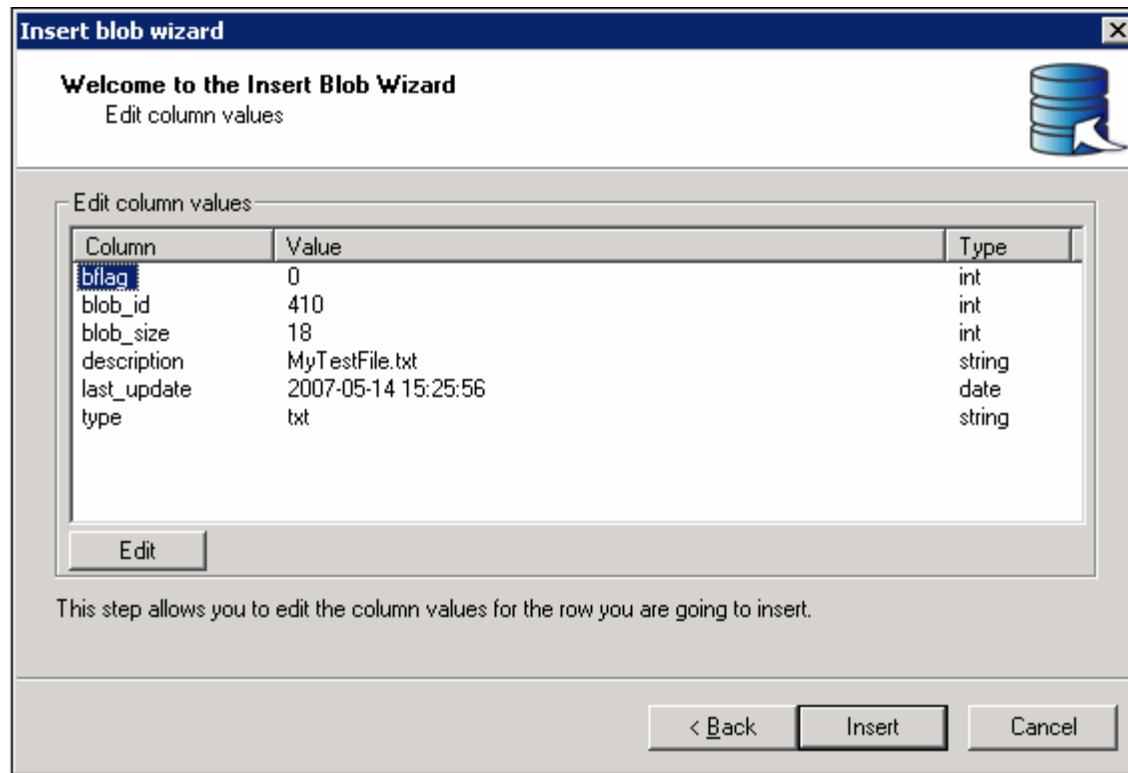
2. Do as follows:

1. Right-click on a table node

2. Select **All tasks** and **Insert Blob** to start the **Insert Blob Wizard**.
The **Insert Blob Wizard** dialog appears.

3. Browse for the file you want to insert and click **Next**.

The following dialog displays columns and values:



You can now modify column values - if desired.

4. Click **Next** to insert the row.

Update blobs

To update a blob, you do as follows:

1. Locate the **Blob Viewer** in **Agresso Management Console**, and then
 1. Select a blob table in the left pane,
 2. Right-click on a blob node in the right pane, and
 3. Select **Update blob**.

A standard file dialog will prompt you to select the new blob file.

2. Select the new blob file and click **Open**.

The blob will be updated.

EXPORT AND IMPORT BLOBS

Zip files

We have implemented Import/Export functionality for blobs packed in zip-files.

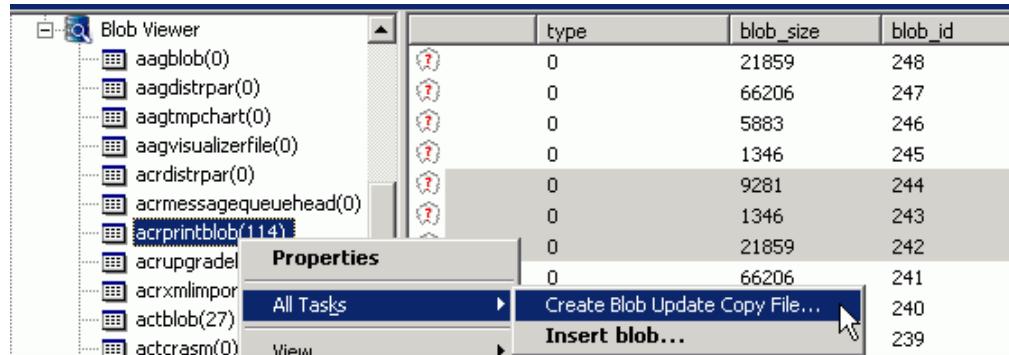
Export a blob

To export blobs from the blob viewer, you do as follows:

1. Open the **Blob Viewer**, select the blob table (left pane) and then the blobs (right pane) you want to export (hold down shift or ctrl to do a multiple selection)

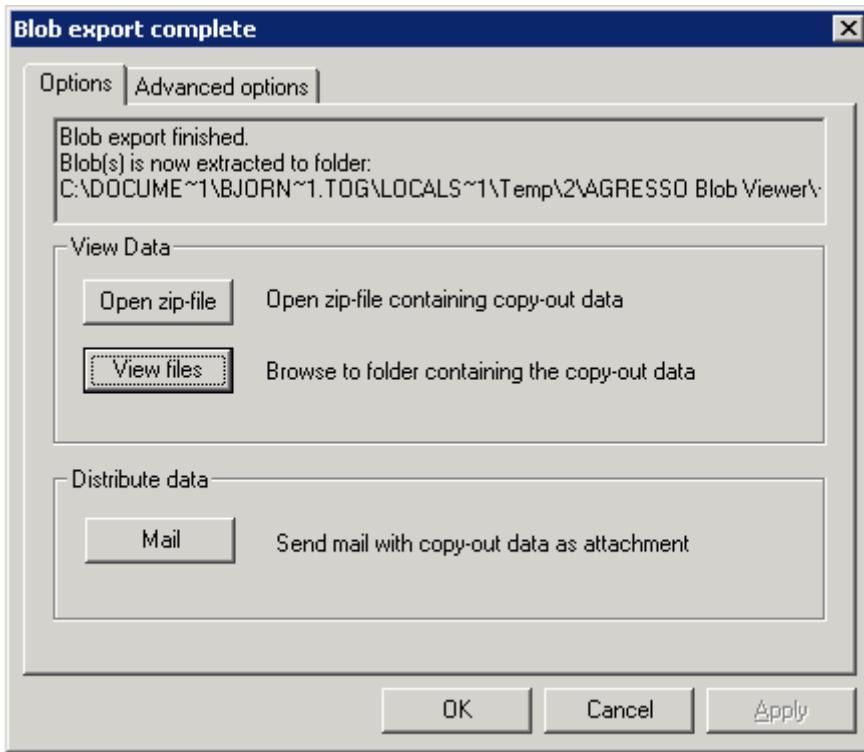
2. Do as follows:

- Right click the blob table node in the left pane



- Select **All Tasks/Create Blob Update Copy File...**
- Confirm your blob selection.

Result: The blobs will be copied to a temp folder and zipped, and you will be given the following options:



The options are:

- Open zip-file:** Opens the zip-file in **WinZip** or other associated archive tool.
- View Files:** Opens the folder where the zip file and the copy-out data files are stored.
- Mail:** Opens the **Send Mail** dialog with the zip-file attached using the AMC associated mail system. If you do not have AMC mail configured, you will be prompted for mail configuration.

Note: If you have trouble with the Mail options, select View Files and right click the zip file to send to mail recipient - this is just as quick.

Import a blob

When you receive a blob update zip-file, you do as follows:

1. Right-click the **Blob Viewer** node and select **All Tasks\Import Blob Updates**.

A standard file dialog allows you to browse for the zip file.

2. Locate the zip file and select **Open**, then select **Yes** in the **Confirm Copy In** dialog.

If all goes well, you get the following receipt: **Blob(s) is now successfully imported**

Server Configuration

CREATING A DATA SOURCE

Prerequisite

The data source node in **AMC** is used as an entry point for the server installation (which will connect to this data source). AMC will also connect to the database through this data source, to be able to make configuration changes.

In order to get access to information in the data source, you need to log in as a user with administrative rights in Agresso. Basic user rights in Agresso is maintained in the **User Master File** in the Agresso Smart Client, located under **System Administration/Users and Access/User Master File** in the Agresso menu.

User master file (example)

In order to create the data source, you will need administrative rights on the server you are working from. But, in order to connect to the data source through AMC, your Agresso user rights must also be in order!

Creating a data source

Always start with the data source

You always start configuring the server by adding the data source. Then you configure the server components using this data source.

Back Office Data Sources are used by all server components to connect to the Agresso database.

The following components use the data source to connect:

- Business Server – including the service, the processes, and the reports.
- Centrally Configured Clients (AMC copies the information to the agresso32.ini file available in the shared folder).
- The Self Service client and Web Services.
- AMC.

As part of the procedure below, you have to enter the name (or IP address) of the database server, as well as the name of the database. You therefore need to have this information ready before you start.

Procedure

The following steps describes what you have to do after you have started **Agresso Management Console**:

1. Right-click the **Backoffice Data Sources** node in the left side navigation pane and select **New/Create New Datasource**.

Example

This will launch a wizard to guide you through a few simple selections.

[Backoffice Data Source Wizard Welcome](#)

2. Click **Next** to begin. The wizard will prompt you to enter a name.

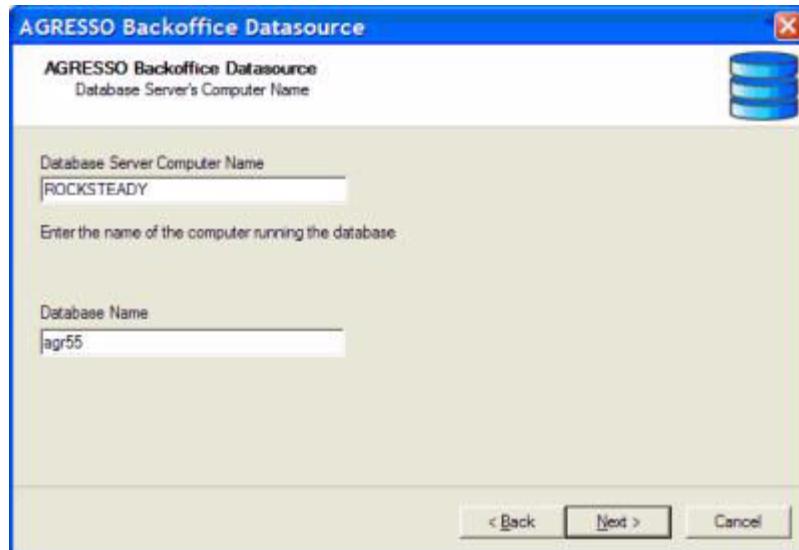
Naming the Data Source

3. Give the data source a meaningful name and click **Next**.

Select Database Type

4. Select the type of database you will connect to and click **Next**.

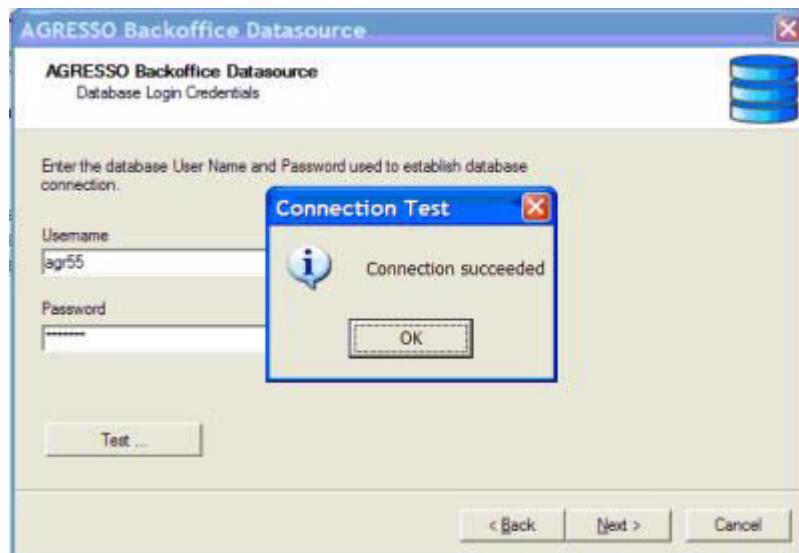
Enter Computer and Database Name



5. Type in the name - or IP address - of the computer hosting the database, and the name of the Agresso database. Then click **Next**.

6. Enter the database user name and password, and click the **Test...** button to verify that everything works. If so, you will get the following receipt:

Connection succeeded



The information is stored in the registry under [HKEY_LOCAL_MACHINE\SOFTWARE\Agresso\Agresso 5.6.1\Data Sources\MyLocalTestDb](#)

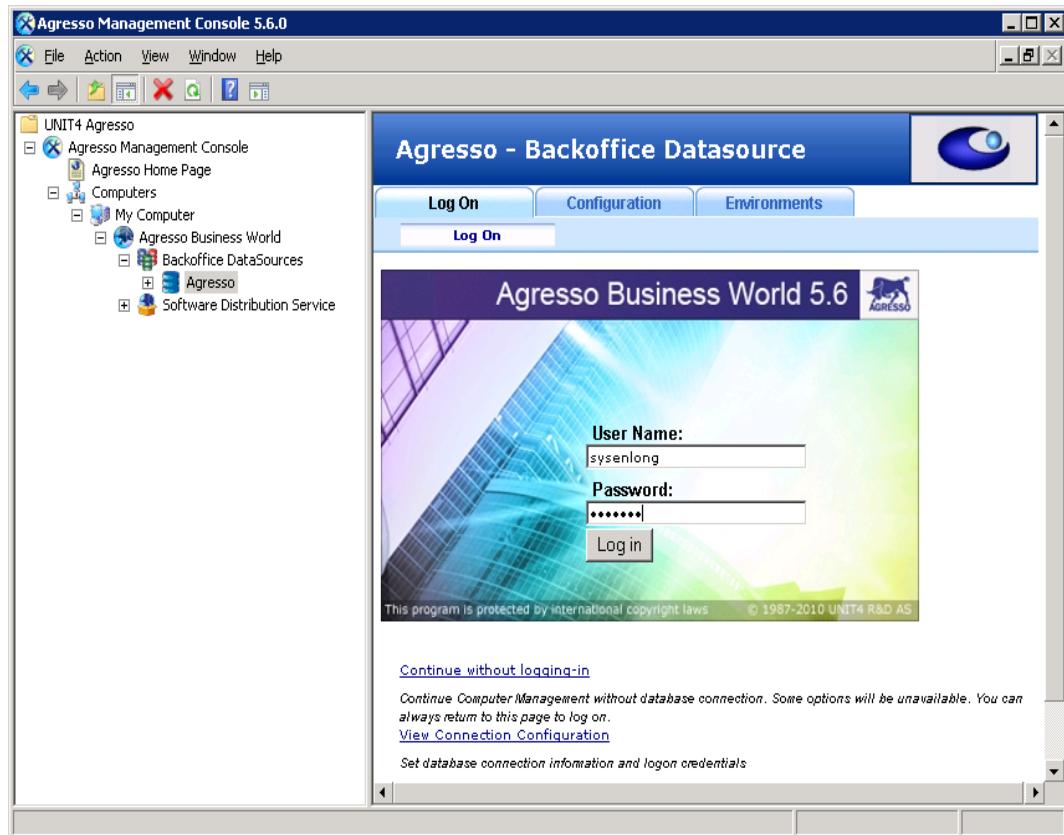
The Data Source dialog in AMC

The Data Source node

You will find the added data source in AMC, under **Backoffice Datasources**. When you open it (right-click) and log in, you will find a dialog with three tabs:

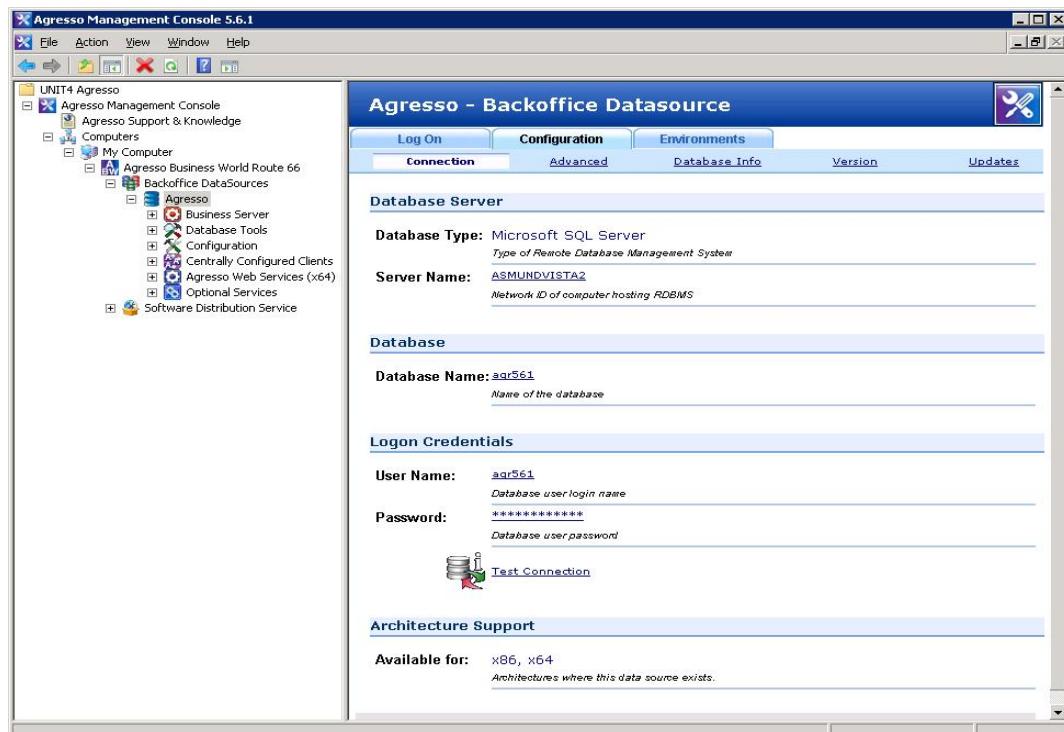
- **Log On** - where you must identify yourself as an *Agresso user with administrative rights* to get full access.

 Example



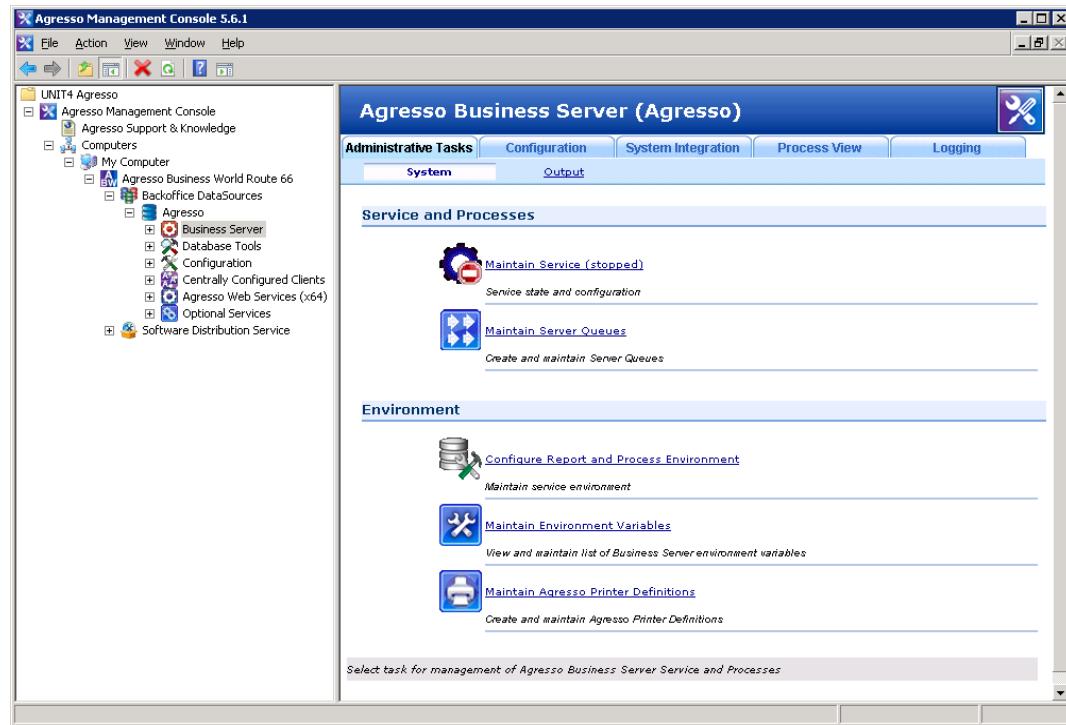
- **Configuration** - where you will maintain the basic database connection information.

 Example



- **Environments** - where you maintain the environment (the components you have selected to install on the server).

 Example



Business Server Environment

INITIALIZE AND MAINTAIN THE BUSINESS SERVER ENVIRONMENT

Prerequisites

You have installed the Business Server or manually configured the computer for server process execution. In **AMC** you will now find a **Business Server** node and **Environment** link.

Initialize the Business Server

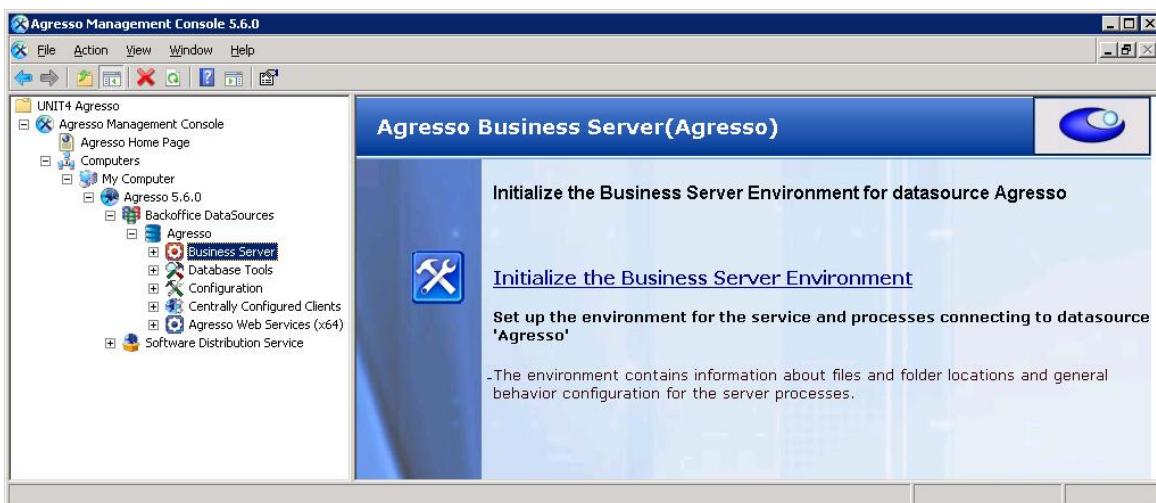
The following procedure describes how you:

- a. Set up the server environment (steps 1 - 5)
- b. Set up rules for the Business Server service (steps 6 - 8).

Set up the server environment

1. Locate and select the **Business Server** node in **Agresso Management Console**. Before the environment is set up, you will only get access to the set up wizard:

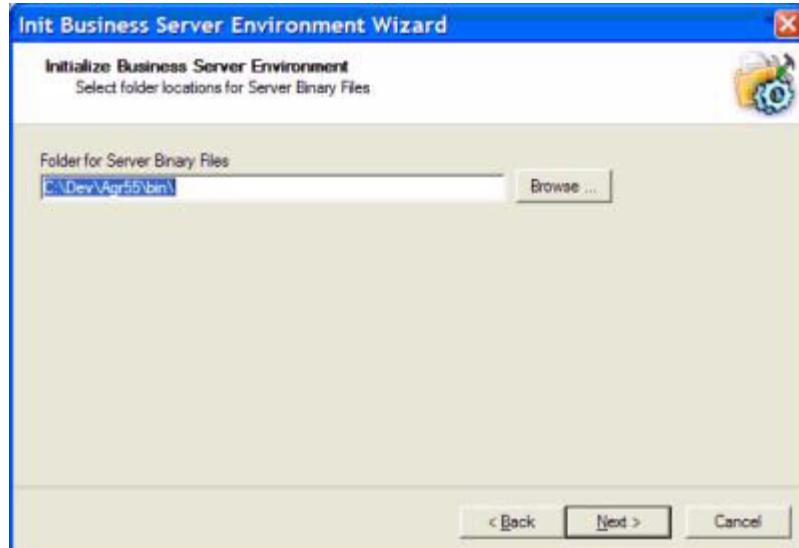
 Initialize the Business Server Environment



2. Click on the [Initialize the Business Server Environment](#) link to start the initialization wizard, and click [Next](#) in the Welcome window.

First, you must enter the folder for the binary files.

Example



3. Select the folder for the Agresso binary files and click [Next](#).

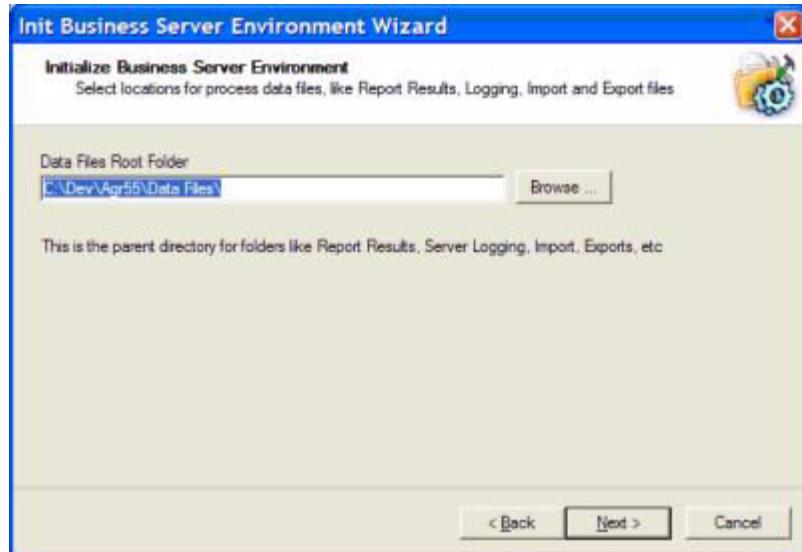
Note: The default folder is stored as environment variable AGRESSO_EXE. The value is based on the registry key [InstallPath](#). Your selection will serve as the base when the rest of the environment variables are set.

Result: The following environment variables will be created, all concerning the location of server files installed by Agresso:

Environment variable	Default path	Description
AGRESSO_EXE	<path>\bin	Folder for Business server binary files (should always be .\bin, so keep the default value)
AGRESSO_REPORT	<path>\Report Writer\	Folder for Agresso report writer and Agresso Excelerator layout files
AGRESSO_COM	<path>\Command files\	Folder for command files used for printing, file handling, etc
AGRESSO_CUSTOM	<path>\Customised Reports\	Folder for customised report layout files
AGRESSO_STYLESHEET	<path>\Stylesheets\	Folder for XML stylesheets

Then you will be prompted to select the root folder for the data files:

 Example



4. Select the data files root folder and click **Next**. The wizard will automatically create a folder with the data source name below this folder, and then all data folders.

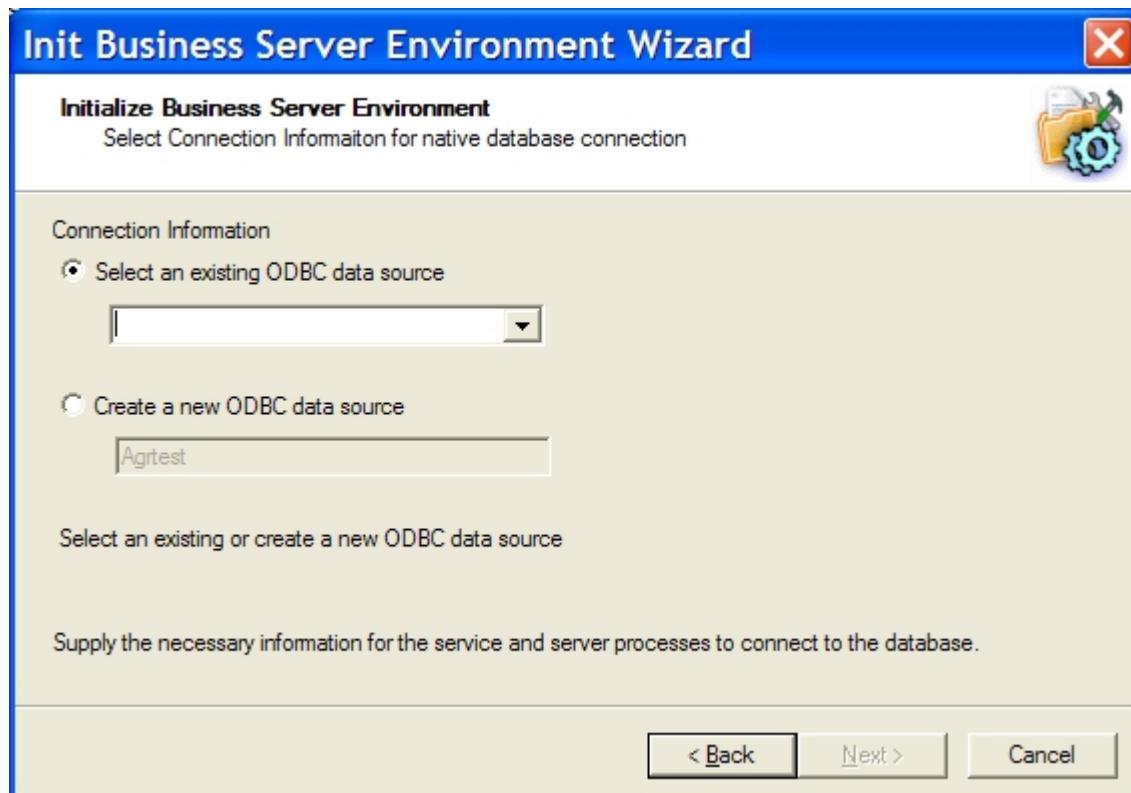
The following environment variables will be created and set to the following locations:

Environment variable	Default path	Description
AGRESSO_PRINT	<path>\<datasource name>\Report Results\	Folder for report output files
AGRESSO_LOG	<path>\<datasource name>\Server Logging\	Folder for report and process logging
AGRESSO_EXPORT	<path>\<datasource name>\Data Export\	Folder for exported data files
AGRESSO_OCR	<path>\<datasource name>\OCR Export\	Folder for OCR export files
AGRESSO_IMPORT	<path>\<datasource name>\Data Import\	Folder for files to be imported into the system

Folders that do not exist will automatically be created.

You will now be prompted to enter connection information:

 Example

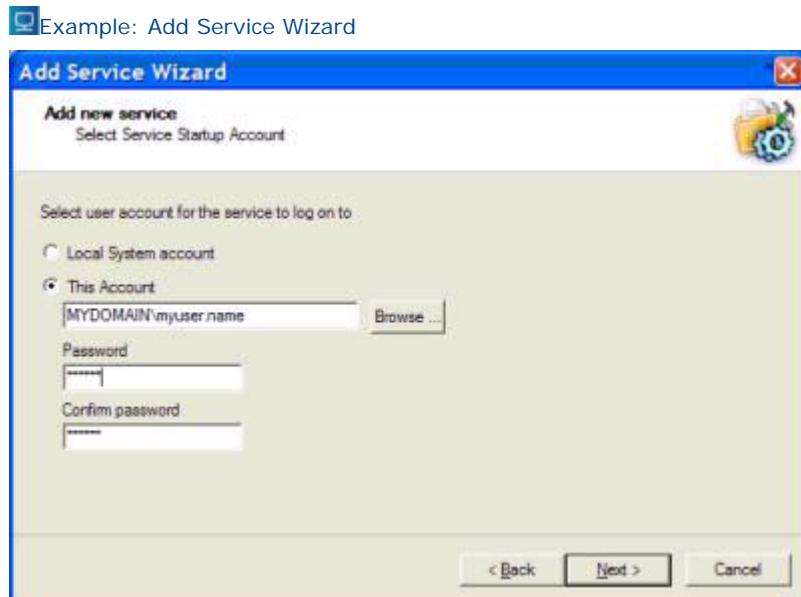


5. If MS SQL database, you must either select an existing ODBC source - or create a new ODBC data source for this purpose. (The Oracle Data Source is fixed.)

Click **Next**.

Install and set up rules for the service

You will enter the **Add New Service** window:



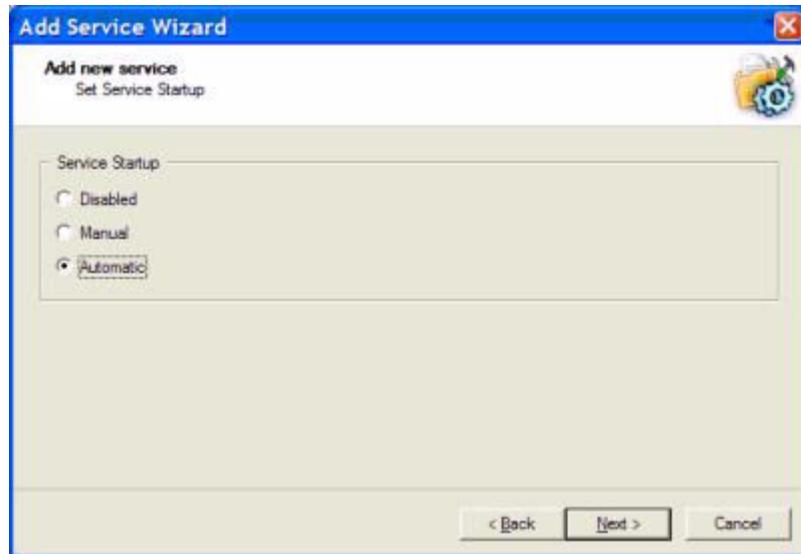
The service is the heart of the Business Server; it schedules and monitors processes, and starts reports on demand of the users.

The service's user account: The service can be configured to run in the context of a local or domain user. Even though the service is able to run under the LocalSystem account, it is recommended that the service is assigned a predefined Agrezzo User Account with the required privileges and resources configured. The Agrezzo User Account has the required access to resources like printers, network folders, registry values, and a mail account for MAPI based mail integrations.

Best practise will be to create a Domain Account named Agresso and add the user to the local Administrator group on the server.

- Fill in the fields as desired, and click **Next**. The **Add new service** window is displayed.

Example: Add new service



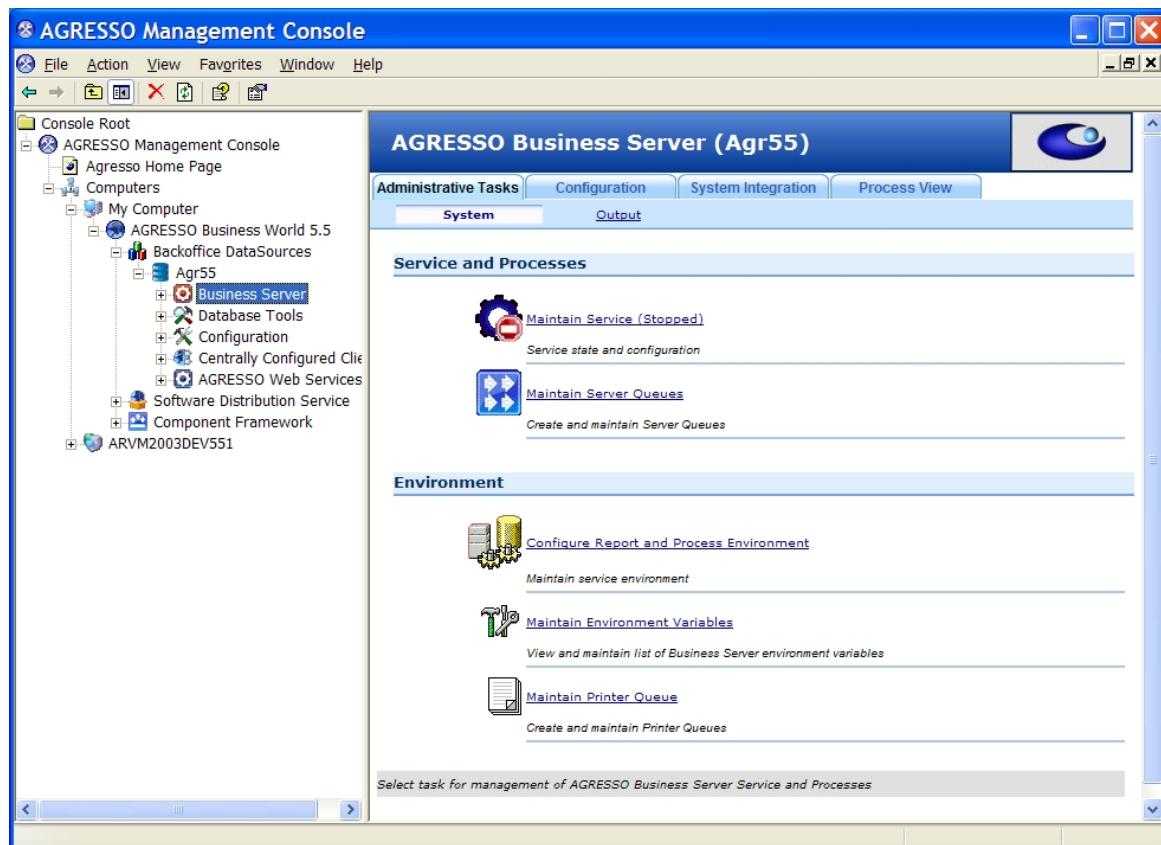
- Select the preferred start up mode and click **Next**.

- Click **Finish** to complete the set up.

Result: The environment will be created and the service will be installed when the wizard is finished.

The **Business Server** node in AMC will now give you access to the functionality you need to administer the server environment.

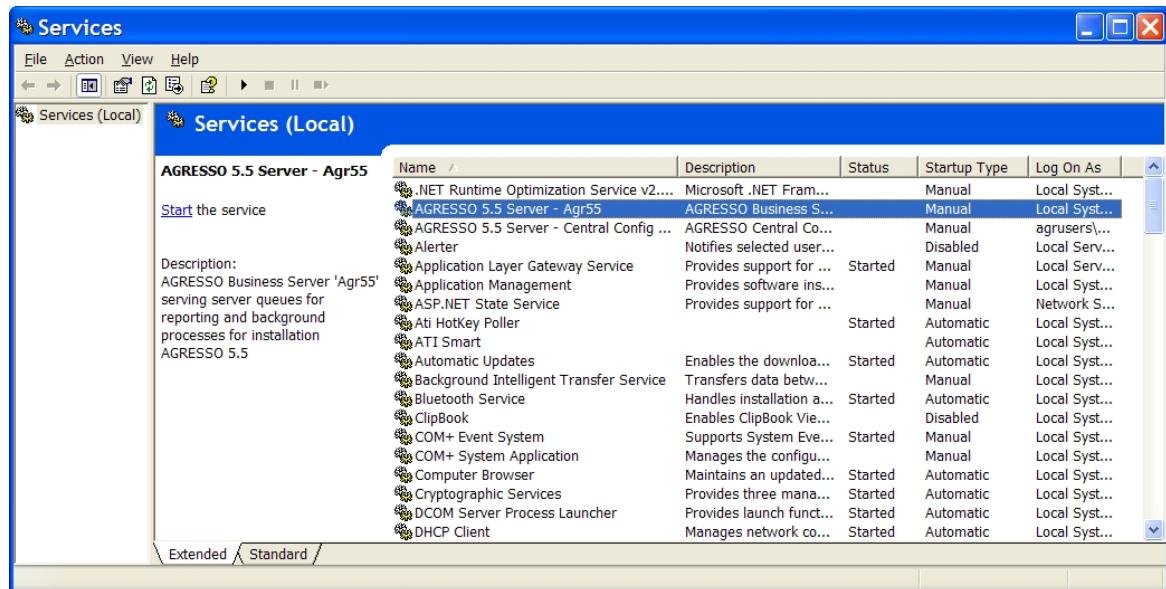
Example: Business Server node



Service maintenance

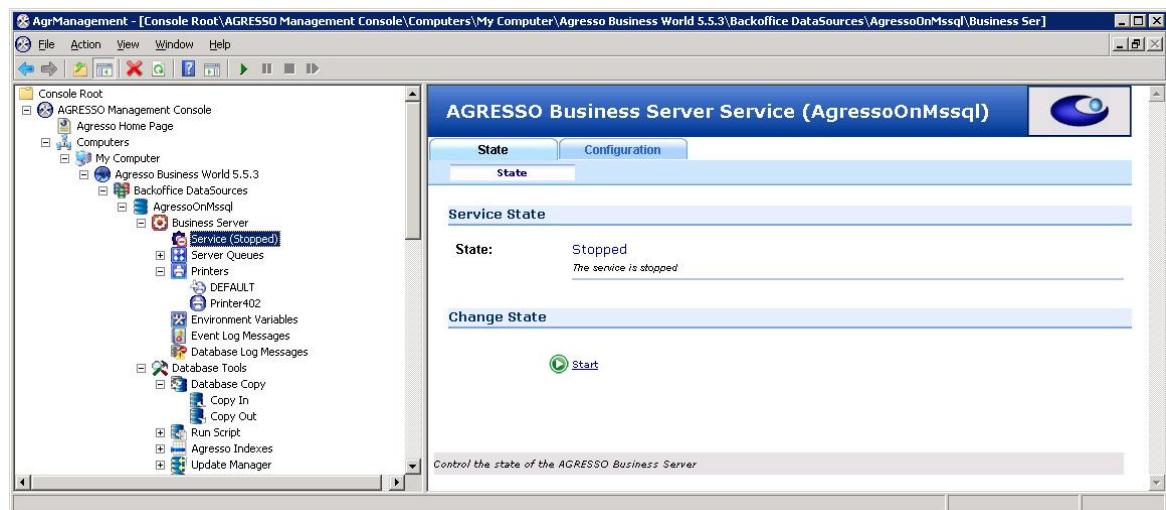
The service is installed as a standard Windows service and can be maintained by the standard Services program in Control Panel/Administrative tools.

Example: Windows Service



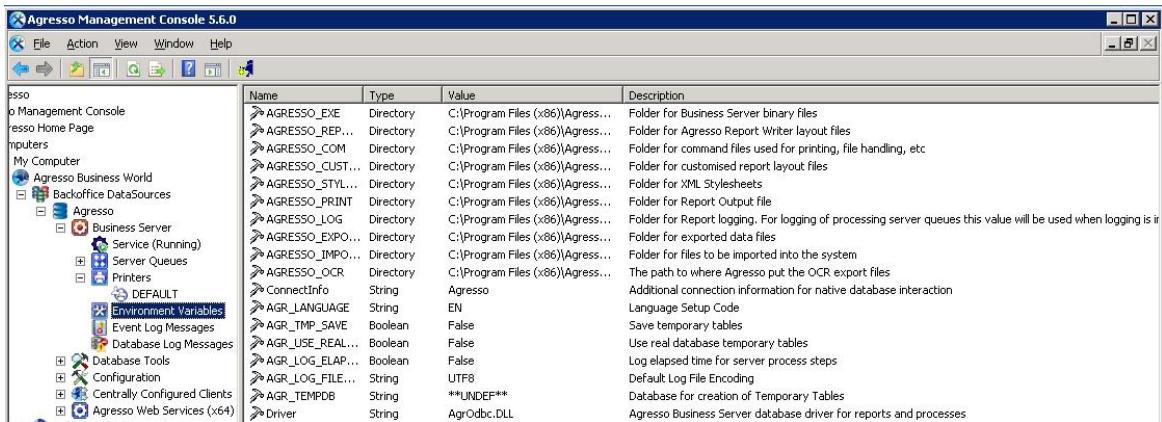
You can also use the **Agresso Management Console**, where you now find the **Service** node after you have opened the **Business Server** node.

Example: AMC Business Server Service



! If you have set the service to log on as a user, you must remember to change the password on the service as well when changing the password on the user account. Otherwise the service will fail to start due to a login error.

Example: Environment Variables



The **Environment Variables** node is used to list all configuration information for the Environment. This gives you quick access to the variables. Double click or right click a variable to enter the **Properties** dialog.

Other environment variables

Environment variable	Value
AGRESSO_SCRATCH	A directory used as a temporary location for reports. By default this parameter is not defined and Windows TMP environment variable will by default be used.
AGR_LANGUAGE	Language setup code
AGR_ALLOW_ERR	Allow errors during batch input(GL07)
AGR_USE_REAL_TABLE	Use real database temporary tables. This is only recommended for trouble-shooting purposes due to performance issues. (it's recommended to configured this setting from the logging configuration page)
AGR_DEBUGGING	Logging level. 4 is the maximum log level (full detailed logging). It's recommended to configure this settings from the logging configuration page
AGR_LOG_FORMAT	Default log file format. If the value is set to 1, xml logging will be used.
AGR_TEMPDB	By default this variable is not defined. This variable must be defined if a separate database should be used for help tables. See also: Agresso On MS SQL Server
AGR_TMP_SAVE	Save temporary tables. Temporary tables will not be deleted when a server job is complete. It's recommended to configure this settings from the logging configuration page.
EXPAND_PATH_WITH_CLIENT	If set to "On", the environment variables AGRESSO_PRINT, AGRESSO_EXPORT, AGRESSO_IMPORT, and AGRESSO_OCR, will be expanded with the client code of the report order if the client code exists. " c:\agresso\Data Files\mydatasource\EN\Report Results " for client EN. If set to "append", the client folder will be the last part of the new path: " c:\agresso\Data Files\mydatasource\Report Results\EN ".

GENERAL LOGGING SETUP

Properties: Basic logging options

You use the **Properties** link on the **Logging** tab (**AMC**, **Business Server** node) to enter values for a few, basic logging parameters.

Setting Logging Options (example)



The screenshot shows the Agresso Management Console interface. On the left is a tree view of the system structure, including 'Agresso Home Page', 'Computers', 'My Computer', 'Agresso Business World 5.6.0', 'Backoffice DataSources', 'Database Tools', 'Database Copy', 'Configuration', 'Print Configuration', 'Mail Configuration', 'SMS Configuration', 'Logging Configuration', 'Licenses', 'Customization', 'Help Configuration', 'Centrally Configured Clients', 'Agresso Web Services (x64)', 'Web Sites', and 'Default Web Site'. The 'Logging' tab is selected in the top navigation bar. The main pane displays the 'Log Properties' configuration page. It includes sections for 'Log Folder' (set to 'D:\Agresso 5.6\Data_Files\agr56test\Server_Logging'), 'Default File Encoding' (set to 'UTF8'), 'Log Time Usage' (checkbox), 'Other' (checkbox for 'Custom process log archiving' set to 2 days), and 'Temporary tables' (checkbox for 'Use Real tables for help-tables' and 'Keep Temporary tables').

Log properties

Field name	Description
Log Folder	The default folder for logging. The value is stored in the environment variable AGRESSO_LOG. When logging is initialized, the AGRESSO_LOG directory will be used by default for log files created by the Business Server. If the AGRESSO_LOG environment variable is modified, the logging configuration will need to be updated to use this directory; this can be achieved by re-initializing logging. The "Server destination" logging destination used by the report queue, will always create log files in the directory defined by AGRESSO_LOG.
Default File Encoding	The value is stored in the environment variable AGR_LOG_FILE_ENCODING. Default value is UTF-8.
Log Time Usage	The log file will show the time each task (query) uses. This might be useful during optimization.

Other

Only relevant if you create your own server queues.

Temporary Tables

Field name	Description
Use Real tables for help tables	Set the server processes to use real database temporary tables. Real temporary tables are not logged by the database and are therefore faster. ! The service will always be logging to the file Agresso Server 5.6-<datasource name>.log. This log is meant for extended history information as well as for troubleshooting purposes.
Keep Temporary tables	A server process creates one or more temporary tables (working tables) during its lifetime. The tables are by default deleted during the process' clean-up phase. Mark this check box if you want to keep the working tables. Use this setting only on special occasions, for instance on request from your support provider. ! This setting should only be used for troubleshooting purposes.

Server Queues

SERVER QUEUES AND PROCESSING SERVERS

Server queues and jobs

A server queue consists of one(!) or more identified *jobs*, ready to run under control of the Agresso Business Server.

A job, in this context, is a series of predefined operations for a set of data. The job's position in the queue determines when it is run, and when the data will be processed. As a result of the job, some Agresso tables may have been updated, a report can have been produced and printed - or both.

Standard Server queues

In Agresso, we differentiate between two types of server queues:

- report queues, and
- process queues.

Report queues

A report queue consists of one or more jobs that are ordered from an Agresso client (by a user). Normally, the user will use report parameters to provide the correct data for a job, and the Business Server will retrieve these data from the database before running the job.

There are two types of report queues:

- **Serial queue:** A queue running one report at a time. This assures that reports ordered in a specific order also will run in the same order.
- **Parallel queue:** A queue that can run two or more reports simultaneously. Time-consuming reports will thereby not completely block other reports, given available time slots.

You must, however, ensure that reports running simultaneously are designed to do so. Use the report/server queue filter available in the Agresso Smart Client to do this.

Example: It is safe to put all reports marked as listing only (not update) into the same parallel report queue.

Process queues

A process queue consists of one or more jobs that are predefined by the Agresso Business Server, and which will be processed by an Agresso Processing Server program.

An Agresso processing server program runs (on the server!) under control of the Agresso Business server, at defined intervals. Normally, a processing server will parse certain tables and check if the necessary conditions for further processing are met. If so, the processing server will run the job - according to the queue settings.

Processing Servers

The Agresso Business Server controls the following Processing servers, each with its own queue settings:

➤ Transaction Processing Server (TPS)

The Transaction Processing Server is a server component configured to perform transaction processing. There should never be more than one TPS running per database. The server runs at configurable intervals.

➤ Logistics Invoice Processing Server (ALGIPS)

The Logistics Invoice Processing Server is a server component configured to perform invoice processing. There should never be more than one ALGIPS running per database. The server runs at configurable intervals.

➤ Logistics Stock Processing Server (ALGSPS)

The Logistics Stock Processing Server is a server component configured to perform stock processing. There should never be more than one ALGSPS running per database. The server runs at configurable intervals.

➤ Data Warehouse Server (DWS)

The Data Warehouse Server is a server that does all collection of transactions and updating of balance tables. There should never be more than one DWS running per database. The server runs at configurable intervals.

➤ IntellAgent Processing Server (AINAPS)

AINAPS is processing all events set up in the IntellAgent module. It will also process the alerts generated by workflow, as well as delay steps set up in the workflow processes. It is possible to set up several AINAPS server queues to share the workload, and to assign events to the different queues.

The execution frequency of AINAPS will determine how often an event can be processed (regardless of the schedule defined on each event, the event can not be processed more frequently than the AINAPS is running), so this needs to be considered when setting the frequency. Normally it should run quite frequently (every 3-5 minutes), so that IntellAgent will be able to rapidly respond to changes.

Note: Workflow alerts will only be affected by the run frequency of the AINAPS if the alerts are aggregated (in the Alert setup screen). If not, the workflow service will start the AINAPS immediately.

Message Server (AMS)

The Message Server (AMS) is responsible for processing all messages in the acrmESSAGEQUEUEHEAD and acrmESSAGEQUEUEDET tables. This includes sending new messages to the interfaces defined in AMC, receiving delivery status for sent messages and posting this back to the tables. It will also re-send messages that previously have failed. Currently, AMS is only used for SMS messages generated by IntellAgent.

AMS should be set up to run frequently (every 3-5 minutes) to ensure that messages are rapidly delivered, and that it frequently will try to resend messages that previously have failed.

The AINAPS will order the AMS directly, and messages from IntellAgent are therefore not affected by the execution frequency, except for retries for failed messages.

RESRATE Server (ARS)

The Agresso RESRATE Server is vital to all HRMS/Payroll/Project customers.

ARS is responsible for producing and maintaining resource rates stored at the lowest possible level (by value reference, by resource, by dates) in the new table ahsresrate.

The purpose is to avoid the heavy workload of processing the rate-hierarchy/rate-setup-logic each time a rate is retrieved. Instead the rates are pre-processed by the RESRATE Server and made directly available on ahsresrate (one query away). The RESRATE server is monitoring all rate-relevant changes made in the system and synchronizing ahsresrate.

ARS should be set up to run frequently (every 3 minutes is suggested) to avoid long delays before changes are processed and rates updated (for example when a resource's pay step is changed in HS01). The ARS will stop immediately if no rows are found in the ahsretrigger table (so the overhead of running the RESRATE server is often only one query against ahsretrigger).

ACRALS Server (AAS)

The ACRALS Server uses the changed data from 'Amendment logging' triggers to update shadow tables, move the data to a history table, and update the import table for IntellAgent (this functionality is not utilized by IntellAgent at the moment).

Workflow

The Workflow service runs when the Business server runs, and manages all the Agresso workflow tasks.

Scheduler

The Scheduler is used to start and stop defined jobs - according to a specified schedule.

Invoice Matching Server

The Invoice Matching Service is a server component to match purchase invoice transactions to purchase orders. Registered invoices without missing goods received are matched by the IMS when receipt is done. There should never be more than one IMS running per database.

Database Logging Service

Reads a message queue and insert rows into table acrlog. Can be used with the Message Queue logging destination. Only enable this server queue if you are going to use the Message Queue for logging.

It's recommended that you only enable the server queues that you need.

Enabling of queues

Before you actively set up the rules for queue handling, the Business server will not be able to handle any report request from a user, and it will not run any server processes. Go to [Queue Setup and Control](#).

Tables

Server queues are stored in the table `aagserverqueue`.

QUEUE SETUP AND CONTROL

Set up options

When you right click on the **Server Queues** node in AMC and select **New**, you will have two options:

Server Queue Example

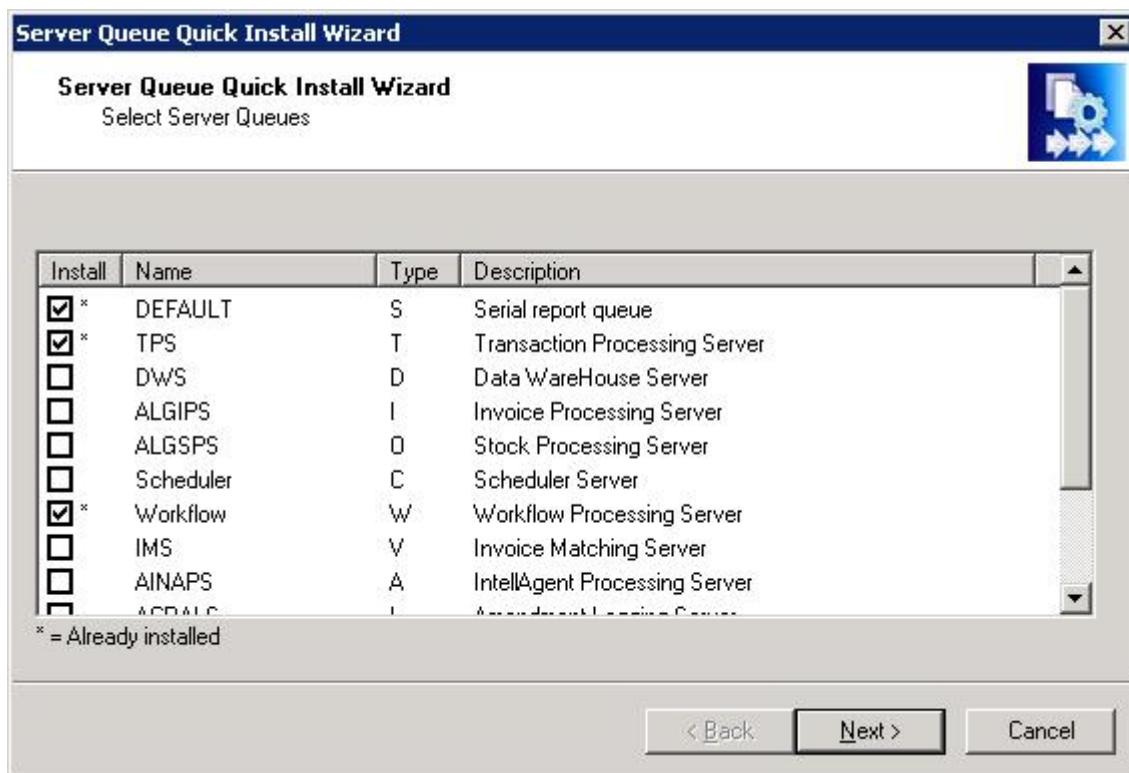


1. You can **Create New Server Queue**
2. You can run the **Quick Setup** wizard to set up the basic queue system.

Quick Setup of a basic queue system

We recommend that you first run the Quick Setup wizard to set up a basic server queue system. By running this wizard, you will set basic rules for all types of queues, by selecting from the predefined list of standard server queues, with default configuration values:

Server Queue Quick Install Wizard



Create New Server Queue

The **Server Queue Quick Install Wizard** allows you to select type, name, number of execution slots, and execution frequency for processing queues, as well as polling frequency for report queues.

When you add (create) a new queue to the Business Server, it will be displayed as a node under the **Server Queues** node in AMC.

Individual configuration of queues

Override default setting

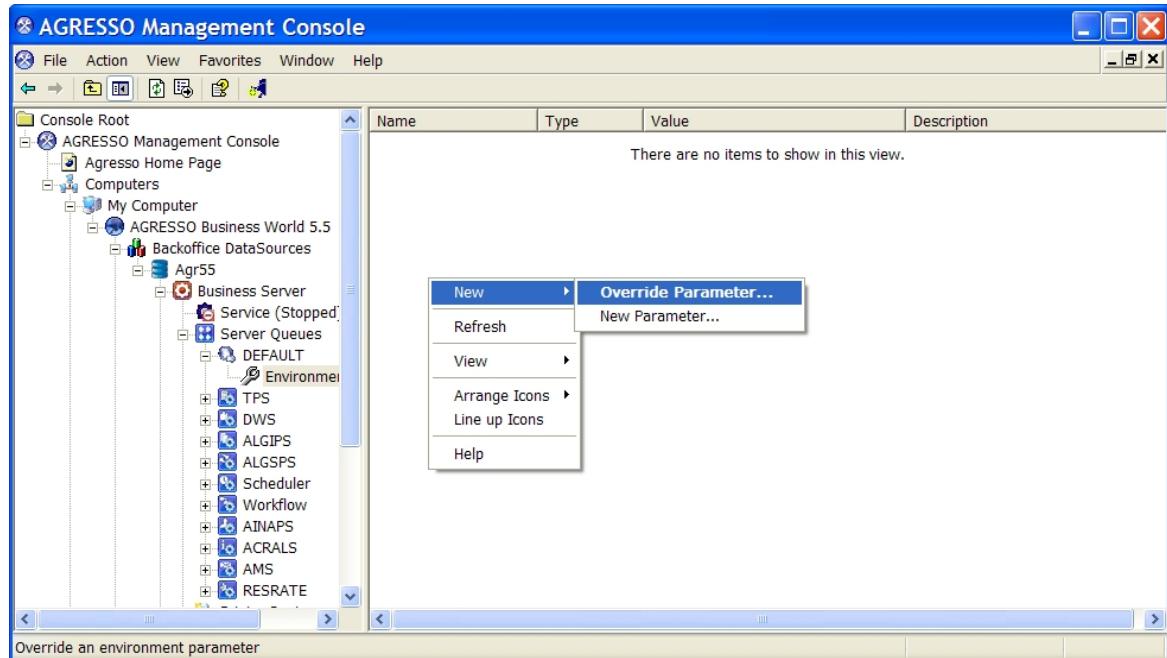
All Business Server environment variables can be overridden for any given Server queue. This allows you to specify individual settings like log options, temporary database for MS SQL Server, or input and output folders to use when reading import files or writing reports. The latter is a requested functionality when you want to prevent certain reports and logs from being available for all users.

Create individual settings for a queue

Before you can individualise a queue, it must be created and available under the **Server Queues** node. Then, you do as follows:

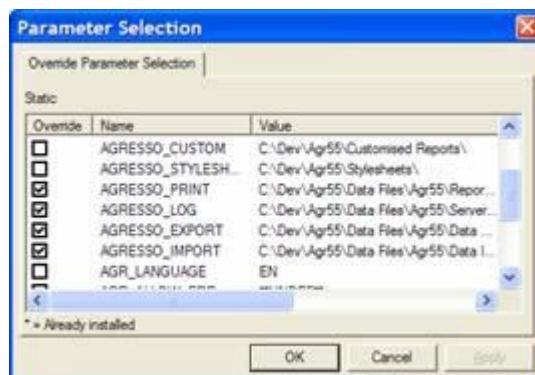
1. Select the **Environment Variables** node for the server queue, right-click the node, and select **New/Override Parameter**. (You can also use **New Parameter** to create a new parameter not yet defined.)
- Note:** In this procedure, we use Override Parameters as an example.

Example: [Override Parameters](#)



Result: The available parameters will be displayed:

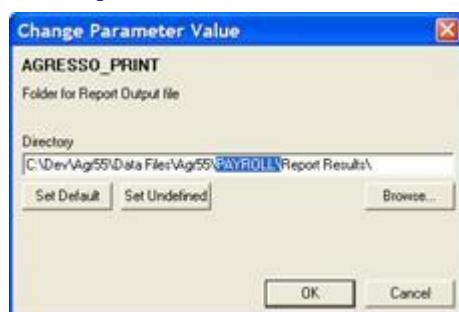
Parameter Selection



2. Select the parameters you want to override and click **OK**. The parameters will be added to the Environment Variables list with the current settings.

3. Select the environment variable and right-click (or double-click) to edit the parameter properties. As a result, the **Change Parameter Value** dialog is displayed.

Change Parameter Value



4. Edit the parameter value and click **OK**.
The queue settings are now modified.

Security considerations

All directories will be created at the server, if they do not already exist. Use explorer to tighten up the security for the new folders to make them inaccessible for non-authorized users.

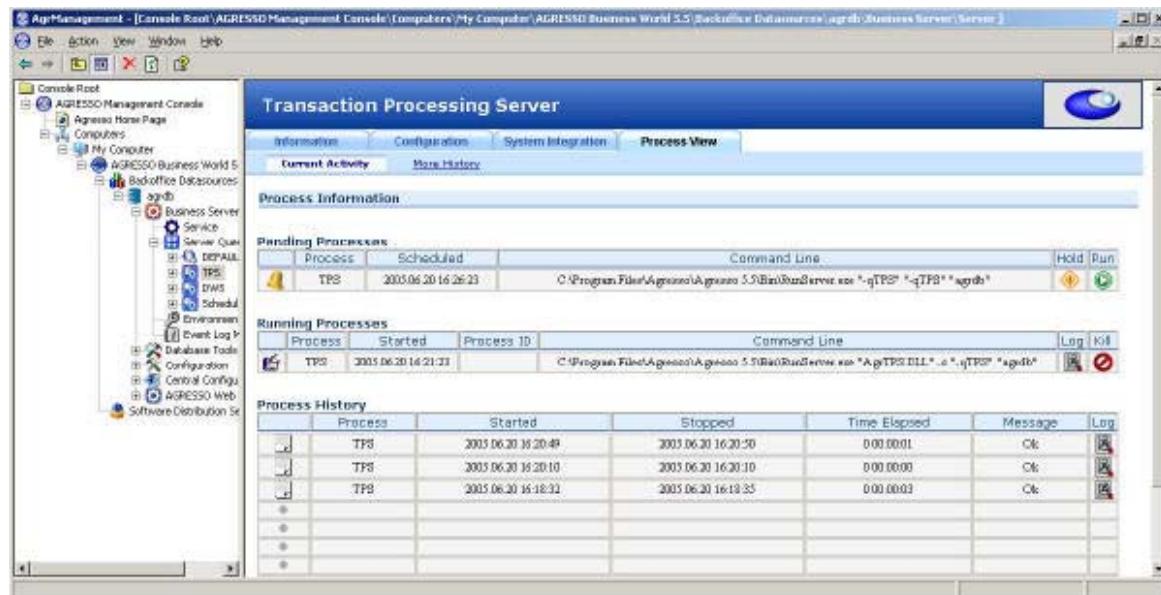
! Remember to keep the write/read access token for the Business Server user.

Queue control - the Process View

When you select a Server queue node in AMC, you will open the **Process View** tab to view and control current queue activity, and view previous events.

There are different actions available for report queues and a processing queues respectively. In the following we give a short description of the various options - for both Report queues and Server queues.

Example: Process View



Pending Reports/Processes

The pending reports (or processes) table is used to view and control reports and processes currently ordered and not yet started.

The following actions are allowed:

For reports	For processes
Hold. Park the report	Hold. Park the process.
Delete. Remove the report from the queue.	Run. Enforce immediate start of the process

Running Reports/Processes

The Running Reports/Processes table lists all currently running reports or processes.

Actions allowed are:

- Log - View Log File.
- Kill - Terminate the report or process. If you are running in the same user context as the Business Server service, you will be allowed to terminate. As a result of this action, the status is set to Terminated (T).

! This is a brutal termination and should be used only when the report or process appears to have trouble terminating in a normal way.

Report History

The **Report History** table is used to view previous executed reports for the server queue.

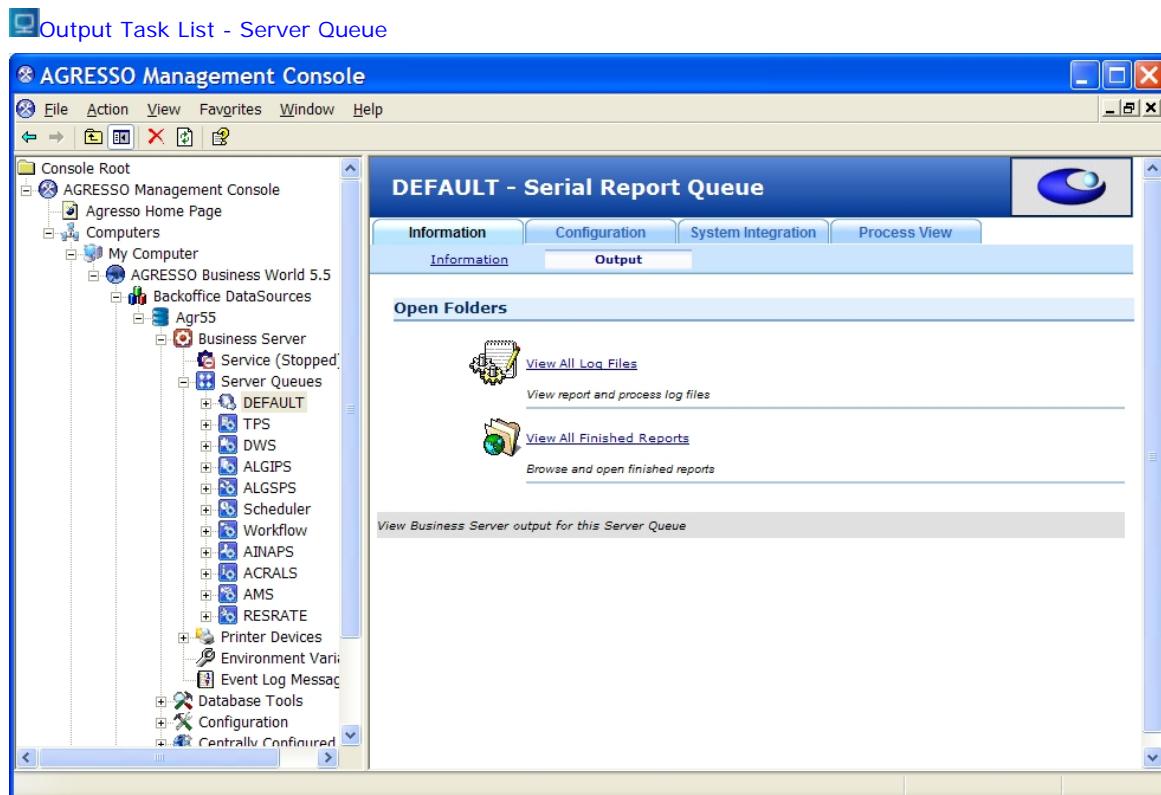
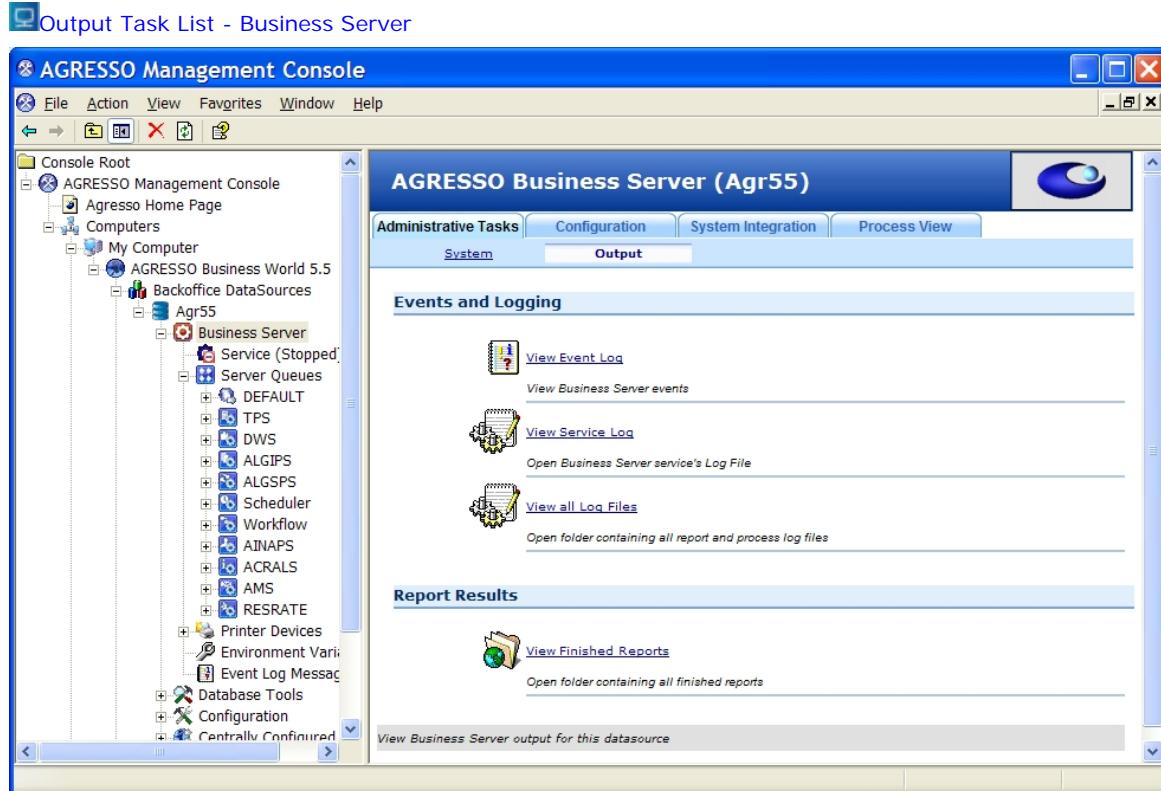
The following actions are allowed:

For reports	For processes
Restart. Will re-run the report.	Log. View log file.
Log	
Output. View the report's output file.	

Quick access to log files

Files and folders used for logging can quickly be localized from the Server Queues Process View or by using the Output task list found in the **Administrative Tasks** menu for both the **Business Server** node and every **Server Queue** node.

Window examples are provided below:



RUNNING BUSINESS SERVER PROCESSES ON MULTIPLE COMPUTERS

Divide the workload

One way to control the processing load for large installations is to divide the server queues on two or more computers.

To set up a second computer, do as follows:

1. Install the Agresso Server software on a second computer and create a Backoffice data source connecting to the Agresso database.

2. Initialize the Business Server Environment on the computer.

The aagserverqueue table: The column `server_name` in `aagserverqueue` holds the name of the server where the queues are supposed to run. The **Server Queues** node in AMC will list all server queues configured for a given database. The name of the remote computer running the queues is found in the column `Host`.

You get an overview over the various computers by opening the **Server Queues** node in AMC.

Example: Server Queues

The screenshot shows the AGRESSO Management Console (AMC) interface. On the left, there is a tree view of the system structure under 'Console Root'. Under 'Computers', 'My Computer' is selected, showing 'AGRESSO Business World 5.5' and 'Agr55'. Under 'Agr55', 'Business Server' is expanded, showing 'Service (Stopped)', 'Server Queues', and other nodes like 'Printer Devices' and 'Database Tools'. The 'Server Queues' node is selected. On the right, a table titled 'Server Queues' displays the following data:

Name	Host	Type	Description
DEFAULT	ASLE	Serial Report Queue	Serial Report Queue
ACRALS	ASLE	Amendment Logging Server	Amendment Logging Server
AINAPS	ASLE	IntellAgent Processing Server	IntellAgent Processing Server
ALGIPS	ASLE	Invoice Processing Server	Invoice Processing Server
ALGPS	ASLE	Stock Processing Server	Stock Processing Server
AMS	ASLE	AGRESSO Message Service	AGRESSO Message Service
DWS	ASLE	Data WareHouse Server	Data WareHouse Server
RESRATE	ASLE	Rate Processing Service	Rate Processing Service
Scheduler	ASLE	Scheduler Server	Scheduler Server
TPS	ROCKSTEADY	Transaction Processing Server	Transaction Processing Server
Workflow	ROCKSTEADY	Workflow Processing Server	Workflow Processing Server

Enable remote configuring of queues

The settings stored in the registry of the remote computer will not be directly configurable. This would have required AMC to hold a remote WMI connection to the host computer.

To configure these remote settings, you can simply add the second computer to AMC:

1. Locate, and then right-click on the **Computers** node in AMC.
2. Select **All Tasks | Add/Remove Computers**

Now you can manage the remote server queues from here.

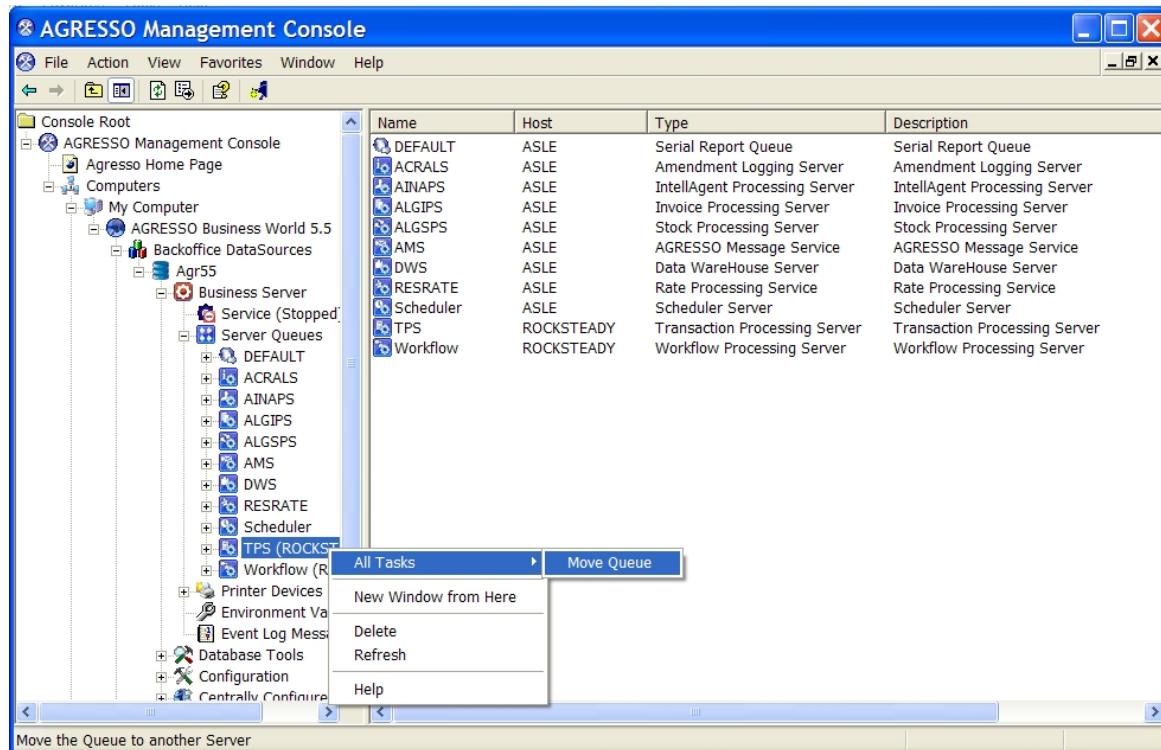
Moving a server queue from one computer to another

! We recommend to stop the services before moving a queue to avoid a situation where two services are serving the same queue, although a service will normally discover that a new queue has been added to the list.

When the Business Server service is started, it always checks the table `aagserverqueue` to find all server queues assigned to it. To move a server queue from one computer to another, follow this procedure:

1. Right-click on the server queue node you want to move (in AMC) and select **All Tasks | Move Queue**.

Example: Move Queue



This will open the **Server Name** dialog.

Example: Server Name



2. Enter the name of the computer you want the queue to be moved to and click **OK**.

Note: This will not move any of the locally overridden environment variables, as they might be dependent of local folder locations and configuration. If the queue requires individual settings, these must be re-configured at the new host computer after the move is completed.

PROCESS EXECUTION CONTROL

Monitoring options

You can monitor all server processes (server queues) of special interest, by using the *Instrumentation* options in AMC. Here you set up actions to be performed when irregularities are detected:

- You can identify potential problems (possible process hang) by deciding a maximum time limit before the process status shall be changed to 'suspicious'.
- You can identify real problems by setting a maximum time limit for when the process status shall be set to 'hanging'.
- You can decide to terminate a hanging process.

Logging to Windows event log

When a process is set up with instrumentation, any changes in the process status will be written to the windows event log. They will all be marked as Agresso events, and identified by the Event Source Name set up for the process. (See [Logging scenarios](#)).

The following Ids are currently in use:

2039 - Suspicious behaviour is detected.

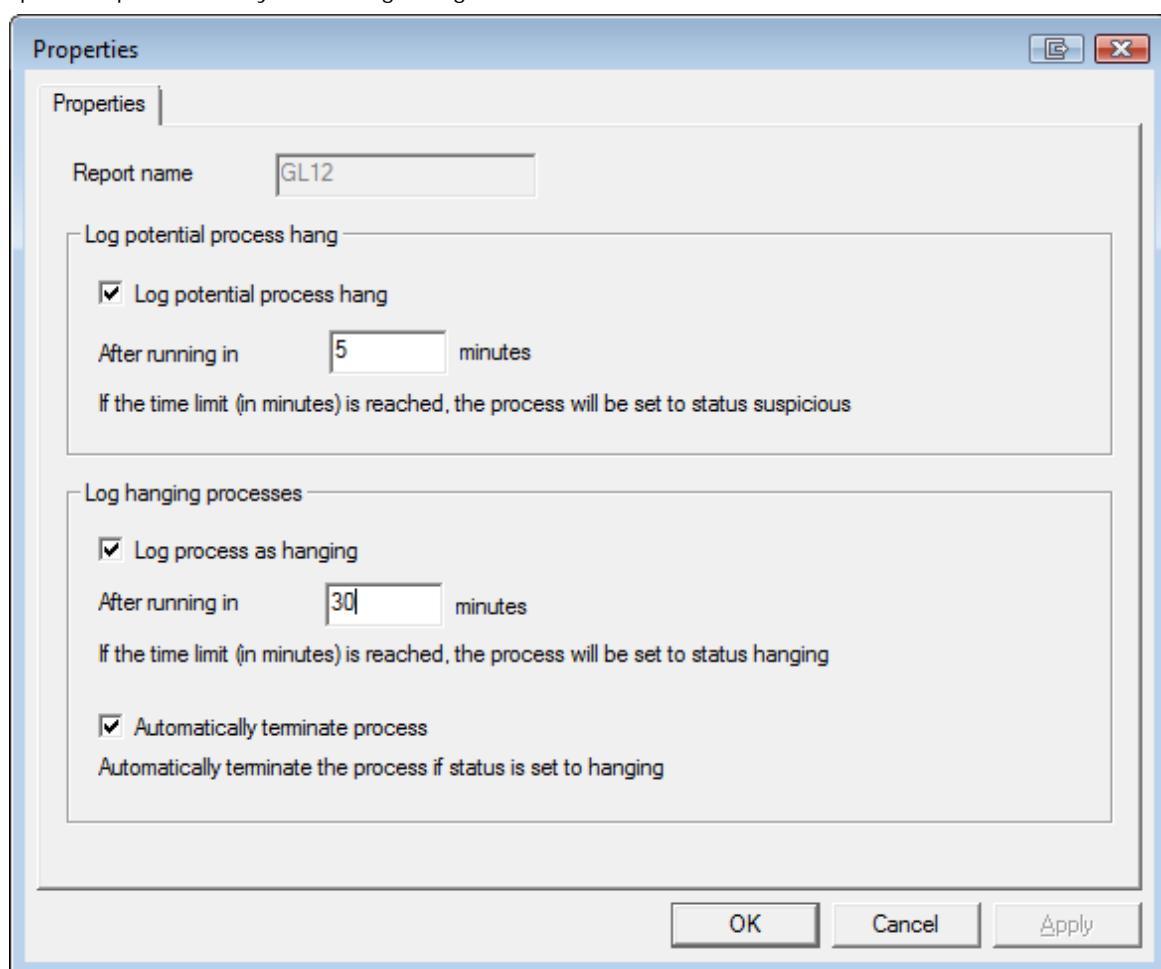
2040 - Hang detected
 2042 - Process is terminated.
 You can get an overview of the defined Agresso events and their Ids by following the link to [Agresso EventIDs](#).

Monitoring a Report

All report queues (initially the DEFAULT queue is the only available) in AMC have an Instrumentation node which allows you to set up monitoring for the various reports:

Report name...	Log potential hang	After(min)	Log hanging	After(min)	Auto terminate
WF01	Disabled		Disabled		Disabled
GL12	Enabled	5	Enabled	30	Disabled
GL27	Disabled		Disabled		Disabled
SO13	Disabled		Disabled		Disabled

You can right-click the **Instrumentation** node to a new report to execution control, and you can right-click a specific report to modify an existing configuration:



Monitoring a Server process

To set up instrumentation for a server process, (e.g. TPS) you find the Instrumentation options behind the Configuration tab (appears when you click the server process node in AMC):

The screenshot shows the 'TPS - Transaction Processing Server' configuration interface. The top navigation bar includes tabs for Information, Configuration, System Integration, Process View, and Logging, with sub-tabs for Processing Server and Instrumentation. The main content area has sections for 'Log potential process hang' and 'Log hanging processes'. Under 'Log potential process hang', there is a checkbox 'Log potential process hang:' followed by a checked checkbox and a text input field containing '5'. A note below says 'If the time limit (in minutes) is reached, the process will be set to status suspicious'. Under 'Log hanging processes', there is a checkbox 'Log process as hanging:' followed by an unchecked checkbox and a text input field containing '10'. A note below says 'If the time limit (in minutes) is reached, the process will be set to status hanging'. There is also a checkbox 'Automatically terminate process:' followed by a checked checkbox and a note 'Automatically terminate the process if status is set to hanging'. At the bottom, a note reads 'Configure instrumentation to allow automatic process control in case of fault. The process control will also log information, warning, and error entries to the Event Log'.

Printers and Printing

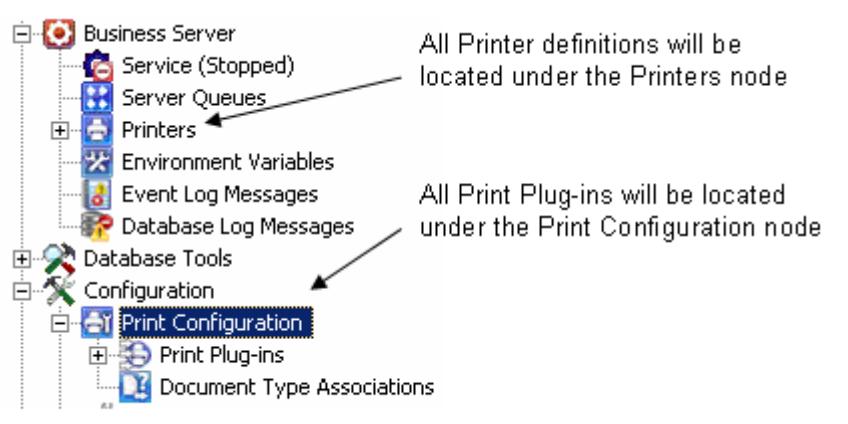
PRINTER SETUP IN AGRESSO

New basic solution

From version 5.5 SP2, Agresso offers a flexible and consistent solution for printer setup and configuration through the **Agresso Management Console**. Old printer definitions can still be configured using the **AG12** screen (**Printers**) in the Smart Client, but you can also use AMC to upgrade your old definitions, and thereby integrate them into the new solution.

AMC nodes

You find the functionality under two new nodes in AMC:



Printer Definitions

The **Printers** node is the parent node for all Printer definitions. A Printer definition can either refer to an existing network printer (Windows printer), or to a program (external print handler) developed for special print purposes. Or it can simply refer to the users default, local print setup.

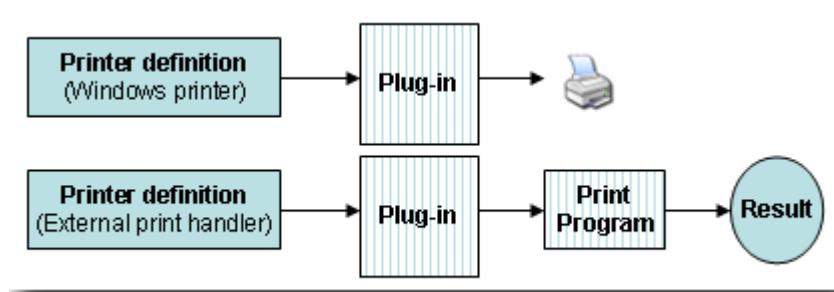
Windows printer: A Windows printer definition will automatically collect printer properties (paper size, orientation, etcetera) from available printers, and you can set property values directly from AMC.

External print handler: An external print handler takes care of special print operations (normally printing that goes to other devices than a windows printer), and is basically a manager for an external program that performs the print job.

You can have several printer definitions using the same available printer or print handler. By tailoring the printer definitions' properties (parameters), you can fine tune the output for different report types.

Print Plug-in

An Agresso printer definition uses a Print plug-in as the means to deliver the print result. The print result is based on Agresso output data and administered by a printer definition. When a report is printed, the report generator identifies the attached printer definition, which in turn activates the attached plug-in to produce the result:



Plug-ins and report types

A plug-in will at least be able to handle one report type, where a report type – also called document type – is identified by the report file extension., such as LIS, XLS an so on.

A printer definition intended to support several report types (for example a windows printer), may therefore be associated with several plug-ins, one for each report type. When a report of a certain type is printed, the plug-in set up to support this report type will take control.

Default plug-in for report type: If you introduce a new plug-in which supports a report type already handled by another (standard Agresso) plug-in, you must set a default plug-in for the report type(s) in question. You can override this default setting by choosing another plug-in on printer definition level.

Custom plug-ins

The use of plug-ins allows us to add print functionality when needed, independent of official ABW releases. When a new, customised report requires special print handling, you can develop a plug-in to produce the print result based on the report data, and add a new printer definition – using this plug-in – as the printer definition for the new report type.

Note: Currently, you will need Agresso assistance to develop new plug-ins.

Upgrading existing printer definitions

You can easily upgrade the existing definitions from AMC. When upgrading a definition for a Windows printer, the printer properties (paper size, orientation etcetera) can immediately be (re-) set from AMC. When upgrading an external print handler (command file), you will automatically get the correct parameters set in AMC.

Note: **AG12** is still available, but should no longer be used for printer definitions.

New table for print configuration data

Printer definitions are stored in the table [aagprintdef](#). When you create Printer definitions from AMC, or upgrade an existing definition, data will be added to the (new) table [aagsystemconfig](#). A printer definition not found in [aagsystemconfig](#) will be treated as *old* (pre-5.5.3).

Report formats

A report format is always part of a printer definition and is identified by the number of columns and rows per page.

Standard Agresso formats

By default, all printer definitions (including external print handlers) are set up to handle all the standard Agresso formats. By unchecking a report format, you exclude reports using this format from the printer definition:

Report Formating for Printer Definition

Report Formats

	Print	Columns	Rows	Report Flags	Form
●	<input checked="" type="checkbox"/>	80	66		default
●	<input checked="" type="checkbox"/>	132	66		default
●	<input checked="" type="checkbox"/>	146	66		default
●	<input checked="" type="checkbox"/>	186	66		default

 [Add User Defined Report Format](#)

User defined report formats

AMC allows you to create custom report formats, which can be required for custom report definitions:

Report Formating for Printer Definition

Report Formats

	Print	Columns	Rows	Report Flags	Form
✗	<input checked="" type="checkbox"/>	40	66		default
●	<input checked="" type="checkbox"/>	80	66		default
●	<input checked="" type="checkbox"/>	132	66		default
●	<input checked="" type="checkbox"/>	146	66		default
●	<input checked="" type="checkbox"/>	186	66		default

 [Add User Defined Report Format](#)

Note: The red X tells you that this format definition can be deleted.

Report types (document types)

Standard types

Agresso standard installation supports nine different report types, identified as follows:

Report type (extension)	Description
AGM	Agresso message type. An AGM report may be generated when something unexpected occurs - for instance that there are no data to print. Text format.
FPG	The report's front page, i.e. attribute values and parameters used when the report was ordered. Text format.
LIS	The standard Agresso report format. Text format.
LOG	Log files. Text format.
PDF	PDF format, can be read by Adobe Acrobat Reader.
RDF	Report Creator format.
RPT	Crystal Report format.
TXT	Text format.
XLS	Excel format.

Plug-ins and report types

ABW comes with a set of plug-ins, covering all Agresso report types, and with no overlap. The table below gives an overview:

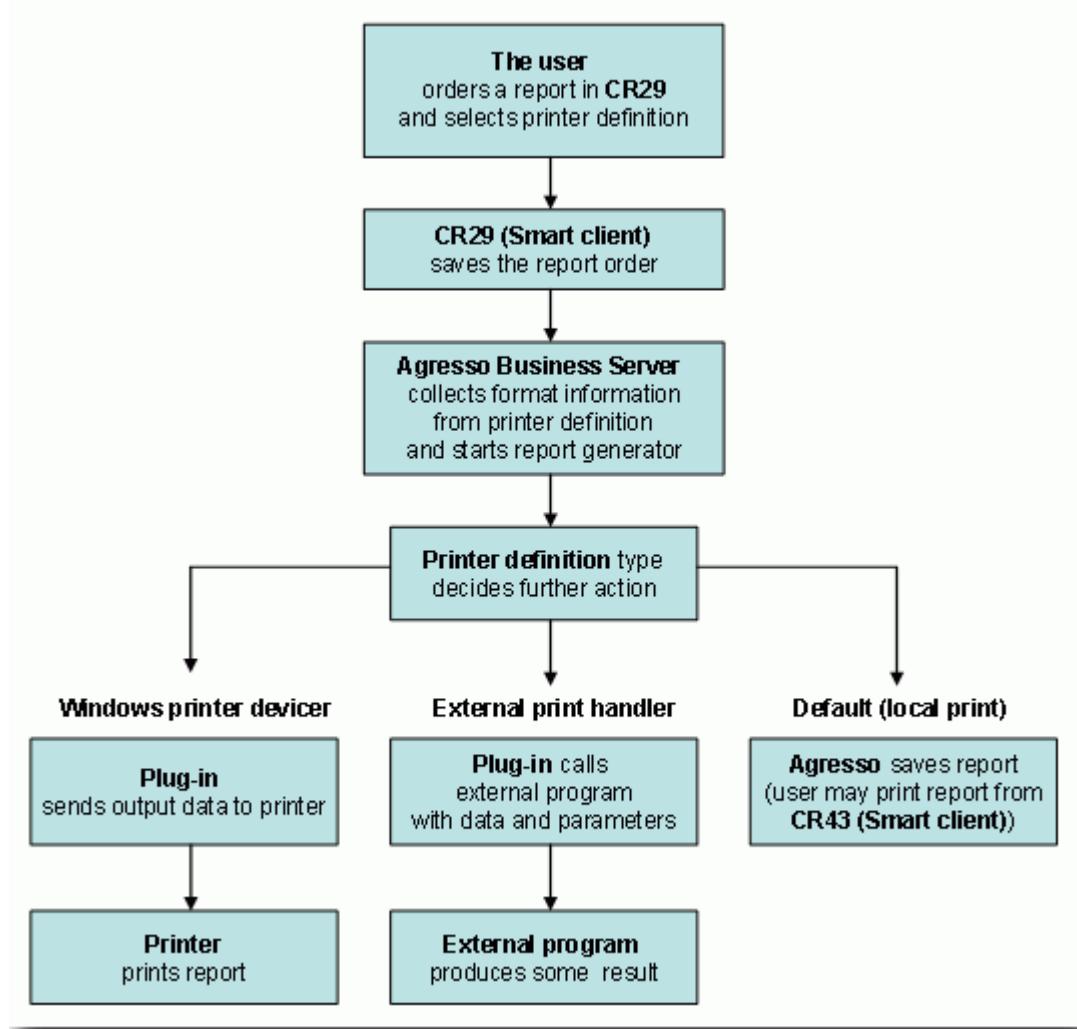
Plug-in	Implementation File	Report (File) Types Supported
---------	---------------------	-------------------------------

Agresso Report Print	AgrPrintReport.dll	*.LIS, *.FPG, *.AGM, *.LOG, *.TXT
Agresso Excelerator Report Print	AgrPrintExcelerator.dll	*.XLS, *.RDF, *.PDF
Agresso Crystal Report Print	AgrPrintCrystal.dll	*.RPT
Agresso External Print Executer	AgrPrintExecuter.dll	All

Support for Crystal Report: The Plug-ins, except **Agresso Crystal Report Print**, are made available in AMC as part of the installation. If you want support for Crystal Report you must add this plug-in manually, after the Crystal Report run-time files are installed.

Summary: Print Process Diagram

The diagram below identifies the main elements involved when an Agresso user orders a report:



SETTING UP PRINTER DEFINITIONS

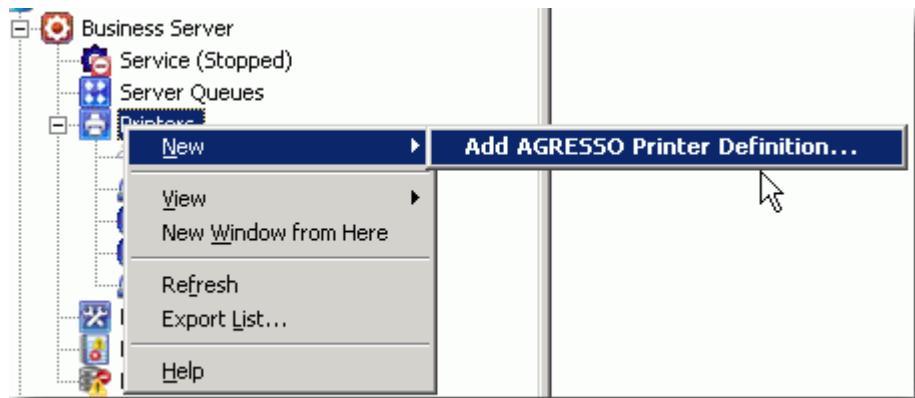
Introduction

This topic covers how you use the AMC to set up and configure the printing options from the Agresso server. See also [Print Configuration Overview](#).

Add a printer definition

After installation, only the (local) DEFAULT printer definition is available. To add a new printer definition, you do as follows:

- Right click on the **Printers** node and selects **New | Add Agresso Printer Definition...**.



This will activate the **Agresso Printer Definition Wizard**.

- Click **Next** and follow the instructions on screen.

Beware of the following:

Windows printer device: If you add a Windows printer, you can only select among printers already connected to the computer running AMC.

External print handler: If you add an external print handler, you will select among plug-ins already registered in AMC. Unless you have added your own plug-ins, the only option will be **AgrPrintExecutor**.

In the current version of AMC, you must also locate the program (executable) to be called from the plug-in to do the actual processing.

Local print: If you add a local printer, you just have to provide a name and description. When a user selects this definition for a report, the user's local PC setup will determine the actual print options.

Setting printer definition properties

When you select a Printer definition, you will get access to the property pages for the definition. The properties are found on two tabs: **Configuration** and **Advanced**. Most properties are self-explanatory, and we will therefore concentrate on the **Advanced** tab, and the option **Print Behavior**.

Print Behavior properties for Windows printers

For a Windows printer, the following properties are available:

Name	Value	Read Only	Hide for user	Not Empty
Front Page	Auto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Name	Value	Read Only	Hide for user	Not Empty
Paper Size	A4 (210 x 297 mm)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paper Source (tray)	Tray 1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Duplex	Open to left	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Colour	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Collate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Print Quality	600x600	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Orientation	Auto	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Front page: The **Front page** property sets the default option that a user of **CR29** will have regarding front page printing. The report will generate a FPG file with report attributes, and handle it according to the selected property value (see table below).

Note: The option **\$PRINTER_SELECT\$** shall not be used in the current version!

The available values are explained below:

Value	Description
Auto	Automatically select how to handle front page based on the Report Definition (recommended).
None	Do not print front page
Print	Always print front page.
On e-mail	Always try to send the front page to the user as e-mail.
\$PRINTER_SELECT\$	Macro identifier. The macro is used by CR29 to populate the drop down list with other Printer definitions for the user to select from. SHALL NOT BE USED in current version.

Device Dependent Settings: The **Device Dependent Settings** properties are taken from the network printer, and allows you to set default printer properties for your definition. You can, for example, have two or more definitions referring to the same physical printer, with different property settings, and intended for different reports.

Print Behavior properties for External Print Handler

For an external print handler, the print behaviour properties offer an unlimited number of possibilities.



Print Execution Configuration: There are three properties related to Print Execution Configuration:

- **Printer Port Name:** Optional. The **Printer Port Name** value should be used together with the **\$PORTNAME\$** macro as parameter (see below).
Note: **Printer Port Name** must not be visible for a user of the Report Order screen (CR29). Always mark it as HIDDEN.
- **Parameters:** The **Parameters** property informs the plug-in how it shall call the associated, external program responsible for the print result. To obtain a degree of flexibility, we deliver a set of pre-defined macros (see below) which can be used as parameters.
Note: The parameter explanations in this section are valid for the default external print handler, and compatible with pre-552 command files. New plug-in may require something else – or nothing at all!
- **Wait for the print operation to finish before continuing.** Needs no explanation.

Parameter format for pre-552 command files

To execute a command file using the fixed parameters as required before Agrezzo 5.5 sp2, use the following command line:

```
"$AGRESSO_COM$$EXECUTABLE" "$FORMNAME$" "$PRINTERQUEUE$" "$COPIES$" "$DOCUMENTNAME$"
"$REPORTNAME$" "$DOCUMENTEXTENTION$" "$DEBUGFLAGS$" "$PORTNAME$"
```

Note on the first macro: Note the following:

- If the command file to run is located in the Agresso Command File folder, use \$AGRESSO_COM\$.
- If the command file to run is located in the Agresso BIN folder, use \$AGRESSO_EXE\$.
- If the command file is placed in any other location, use the full path (e.g. c:\program files\... etcetera).

Available macros

The available macros are described in the table below:

Macro (parameter)	Value
\$AGRESSO_COM\$	The path to the AGRESSO command folder (ends with a backslash)
\$AGRESSO_EXE\$	The path to the AGRESSO bin folder (ends with a backslash)
\$AGRESSO_LOG\$	The path to the AGRESSO log folder (ends with a backslash)
\$AGRESSO_PRINT\$	The path to the Report Output folder (ends with a backslash)
\$COPIESS\$	Number of copies to print. The value is taken from CR29.
\$DATASOURCE\$	Name of the Agresso datasource
\$DEBUGFLAG\$	Debug flag. Value DEBUG if debug level is set to Add Queries or higher, otherwise NODEBUG
\$DEVICE\$	The value from the print_queue column of aagprintdef
\$DOCUMENTEXTENTION\$	Document filename extension, including leading period. This is the extension of the report type, e.g. .LIS.
\$DOCUMENTNAME\$	Name and full path of the document to be printed. E.g. GL12a_4.LIS
\$EXECUTABLE\$	The name (no path) of the third party application or command file to execute (for an external print handler). The value is set on the Configuration tab for the Printer Definition.
\$FORMNAME\$	Formular name – if defined for the report.
\$LOGFILE\$	Name and full path to the log file as defined in the Agresso Log.
\$ORDERNO\$	The order number of the report.
\$PORTNAME\$	Port name as set in Printer Port Name in Print Execution Configuration.
\$PRINTER\$	Name of the Agresso Printer Definition
\$PRINTERDEST\$	The value from the destination column of aagprintdef
\$PRINTERQUEUE\$	The value from the print_queue column of aagprintdef (as \$DEVICE\$)
\$PROCESSNAME\$	Name of the process executing the application/command file
\$REPORTCOLS\$	The number of columns in the report
\$REPORTNAME\$	The name of the report to print (e.g. GL12)
\$REPORTROWS\$	The number of rows to print per page
\$SERVERQUEUE\$	Name of the server queue executing the report

Important: If you know that the substituted macro value may contain blanks (space characters), the macro must be enclosed in quotes, e.g. "\$DOCUMENTNAME\$". To be on the safe side, you can use quotes for all macros.

TROUBLESHOOTING

Report log

Log file

Printing steps are logged in the report's log file, for example *gl12-4.log*.

Log content

The log tells you:

- the name of the file to print,
- how the front page is handled
- the plug-in that is used
- any critical errors that occur during printing.

Parameter for additional details: You can set the parameter [Add queries to log file](#) to get additional details about the printing steps, as well as a reference to a separate Plug-in log file. This log file's name format is:

<file to print>.printing.log (example: [gl12_4.lis.printing.log](#))

and is stored in the AGRESSO_LOG folder.

AgrPrintExecuter: The external print handler AgrPrintExecuter will also log to an own log file for printing when [Add queries to log file](#) is turned on. It logs the command format as well as the substituted values representing the command to execute.

Common Errors

We have identified and described a few common errors that can occur:

1

Error message: Could not open device '\\FP47OSLO\K4S_Ricoh1035x' for printing

Explanation and solution: The Agresso Business Server is running in the context of a network user not having the printer device attached or as Local System. Windows Printer Devices are stored in the registry per user. A service running within the Local System context would not be able to see such information.

Assure you log on the computer running Agresso Business Server as the Network user the same user the service logs on as. Now, add the Windows Printer Device using the Printers applet in the Control Panel.

If the service logs on as Local System, change this to a network user and configure the printer for this user. It is recommended that an own network user is created for Agresso Business Server for this use.

2

Error message: Printer plug-in 'AgrPrintReport' not a registered plug-in for printing

Explanation and solution: The Print plug-in has been removed after the Printer definition was created.

You must re-create the Printer definition or use [AG12 \(Printers\)](#) in the Smart Client to create the Printer definition if you do not want to use the Print Plug-in.

3

Error message: <application name> is not recognized as an internal or external command, operable program or batch file

Explanation and solution: The parameters (via macros) are incorrect. The folder path may be wrong (you may have used \$AGRESSO_COM\$ instead of \$AGRESSO_EXE\$) or you may have forgotten quotes around macros.

Use the log to get a copy of the command that generated the error. Then try to run the command from a command window and study the error messages.

AGRESSO EVENT SERVER

Description

The Agresso Event Server service keeps track of the report activity controlled by the Agresso Business Server. The Agresso Event Server replaces the Agresso Alert Server web services used in previous Agresso releases.

The service allows the smart client to fetch information about report status without querying the database. The "ordered reports" pane in the smart client is dependent on this service to be able show the updated report order status.

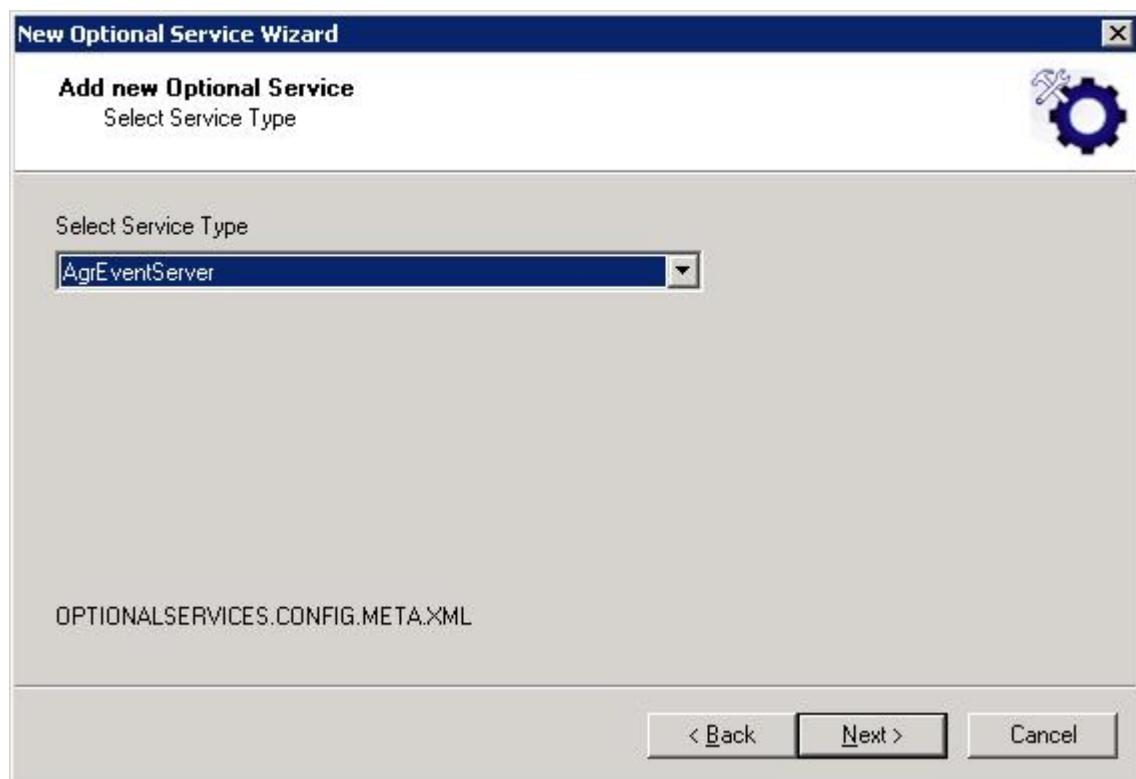
Configuration

The service can be configured from under the optional services node in AMC

1. Right-Click the optional services, select new/Add Optional Service



2. In the wizard, select "AgrEventServer" as service type



3. Click next until the wizard is complete.

Additional information

The smart client use the common parameter ALERSERVER_URL to locate the path to the service, by default the smart client polls the service for changes every 10th second. The poll frequency in the smart client can be configured by adding the REPORT_TIMEOUT common parameter.

A windows firewall exception is added for AgrAlertServer.exe when the service is created.

Client Configuration

CENTRALLY CONFIGURED CLIENTS

Run from network share

Introduction

If you set up the Agresso Smart Client on a network share, you can make it available for as many (local) users as needed, with minimal configuration of the users' computers. The steps are as follows:

1. Create the shared folder structure, with necessary Agresso files.
2. Make sure that the local users get access to the startup file (*Agresso.exe*) - for instance by sending a shortcut through e-mail.

Extended distribution

Normally, you will create the network share on the machine where you installed the Client components and give the users a shortcut to this share.

When necessary, however, you can simply copy the file structure from the original network share to other machines (for instance to a file server in another location), and thus make the Agresso Smart Client available for new user groups.

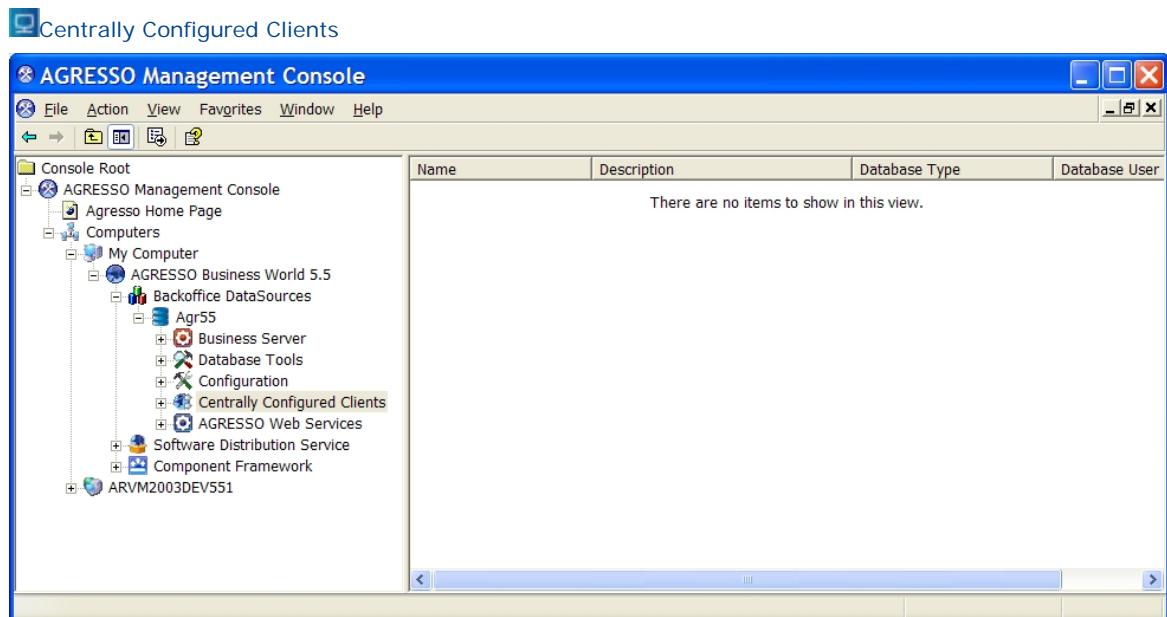
A note on .NET Framework

Please note that all (local) users of the shared client must have the correct .NET Framework version installed.

The Centrally Configured Clients node in AMC

CCC location

To create centrally configured clients, you will use the functionality in the **Centrally Configured Clients** node, found under **Backoffice DataSources/<Agresso Data source name>** in the AMC:



The CCC master installation

All required CCC software was (by default) installed in the bin folder when you installed the Agresso Smart Client on the server. This is referred to as the *master installation* for all centrally configured clients.

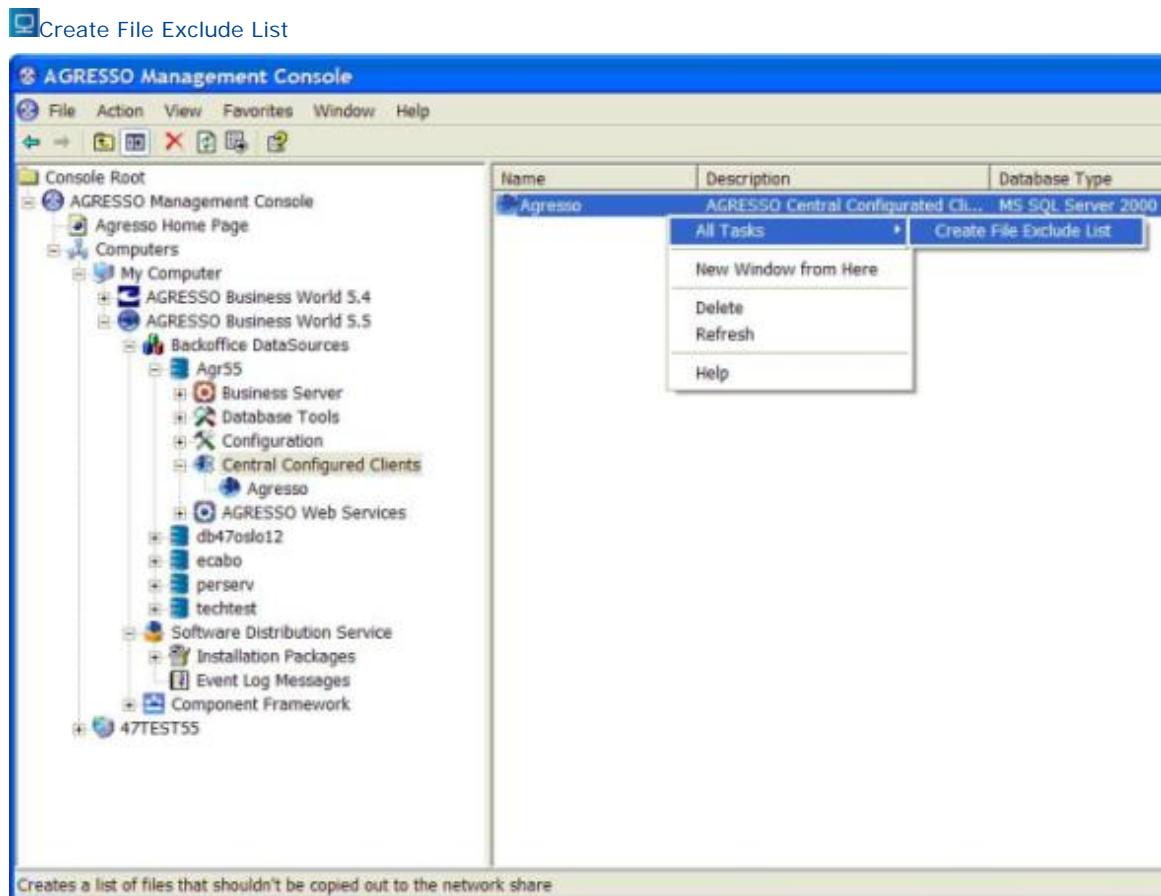
Updates: Service packs are installed on top of the master installation. AMC has functionality to refresh the CCC files based on the master installation. See [Maintaining Centrally Configured Clients](#) for information on how to update existing CCCs with new files.

The File Exclude List

Not all files in the master installation should be distributed to the client users. Some files might be harmful or confusing when executed on the client computers. To prevent files from being copied from the master bin folder to the network share bin folder, you can create a so-called File Exclude List.

Create default file exclude list

You simply right-click the CCC node and select **All Tasks | Create File Exclude List** to create the default file exclude list:



The default exclude list is stored in the file *AgroCCCFileList.xml* in the Config folder. You can use any text editor to edit the file content. The default list will prevent Agresso Management Console, the Business Server service executable, the Copy Applications and the ASQL Application from being copied to the share.

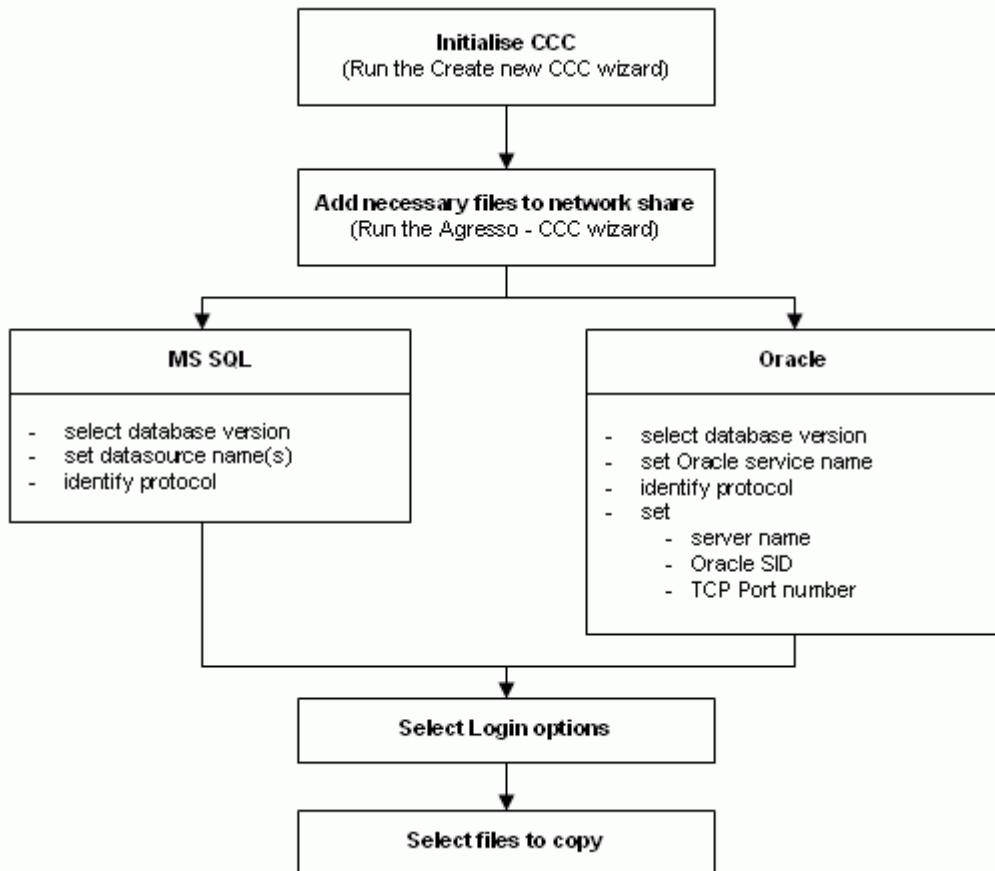
Updates

The content of *AgroCCCFileList.xml* will also be used each time you use the **Update CCC Files Wizard**. If you later manually delete one or more files from the share's bin folder, remember to add the file to the File Exclude List to prevent the file from being copied next time the files are updated.

Central configuration process

Process diagram

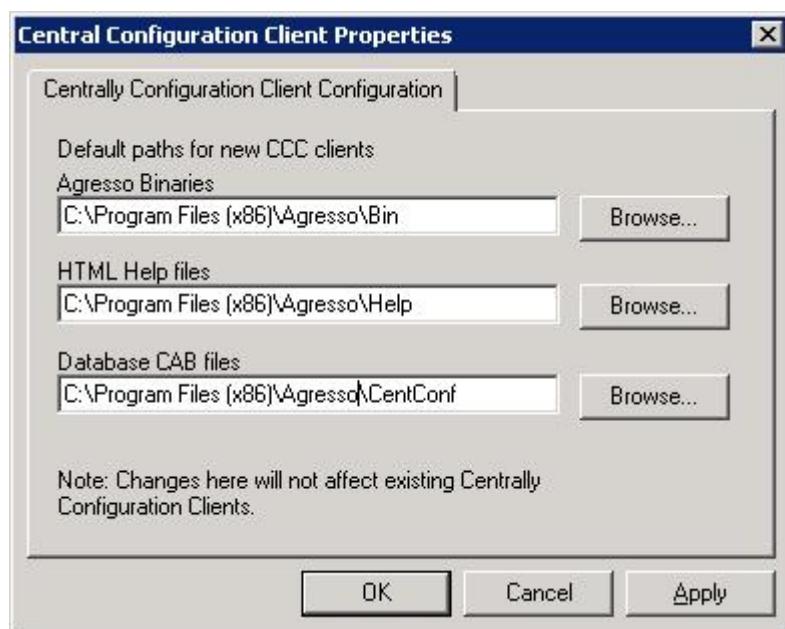
The diagram below outlines the main tasks when setting up centrally configured clients:



File and database information

During the setup, common data source information such as database type, database name, login information etc, is collected from the BackOffice Data Source.

The base paths for both the client files and the database driver files are read from the registry. You can verify and change the paths if they are not correct. To do this, right-click the **Centrally Configured Clients** node, and select **Properties**.



Native connection files

The following versions of the native connection files will be installed in the DB folder at the network share:

- Oracle 11

Packages for CCC users

If you need the Agresso VBA msi package please contact your local Agresso support center.

CREATING A CENTRALLY CONFIGURED CLIENT

Two tasks

To create a centrally configured client, you must:

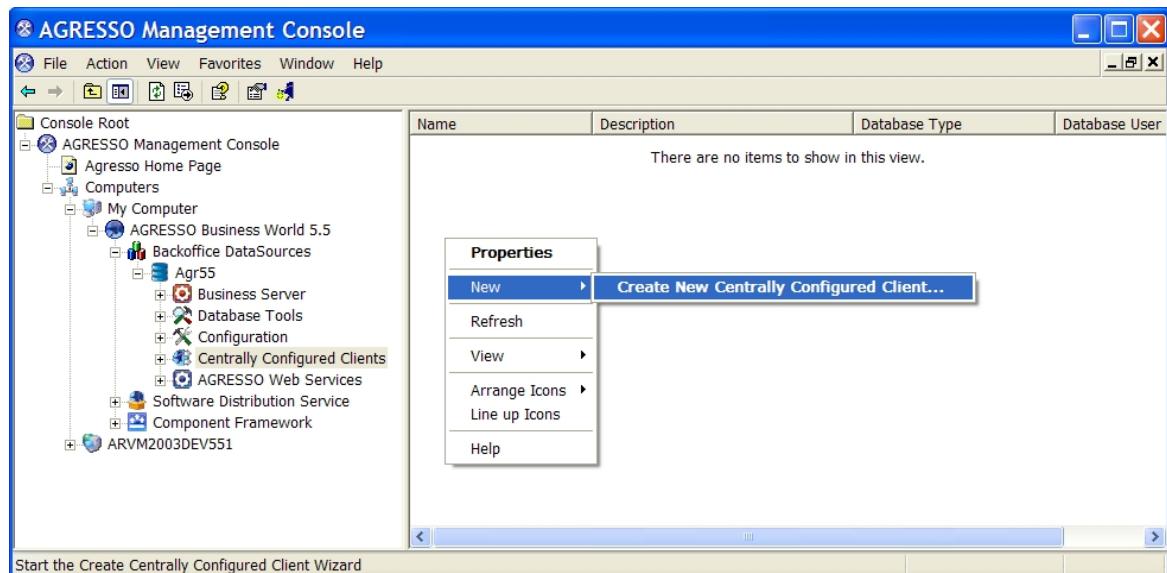
- Create a network share.
- Initialise the CCC by adding files and configuration information.

Create a network share

To create a network share for centrally configured clients, do as follows:

1. Right-click the CCC node and select **New | Create New Centrally Configured Client**.

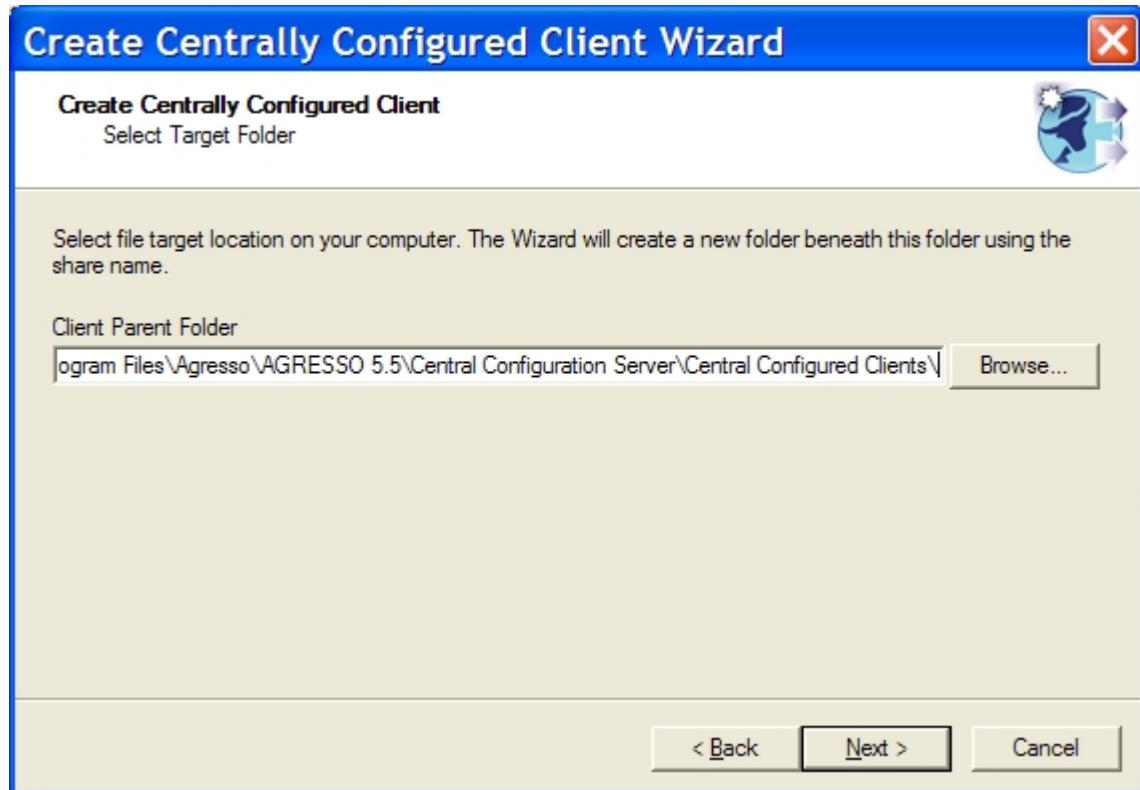
Example: Create New CCC



The CCC Wizard is activated, prompting you for a name of the network share.

2. Enter the name of the network share and click **Next**. You will later find this name as the name of the CCC root folder. This will open the **Select Target Folder** dialog.

Example: Select Target Folder



3. Verify that the parent folder path of the network share is correct and click **Next**.

! *The Browse button is disabled when doing remote configuration, as all file operations are processed at the remote server.*

4. Click **Finish** to complete the wizard.

Initialise the CCC

You now have to initialise the CCC, by adding necessary files and configuration information. This will complete the initial CCC configuration.

! *By default, all files found in the master bin folder will be copied to the share's bin folder. You can exclude files to prevent them from being available to the users. To do this, right click the CCC node before you initialize the CCC. See [The File Exclude List](#) for details.*

Do as follows:

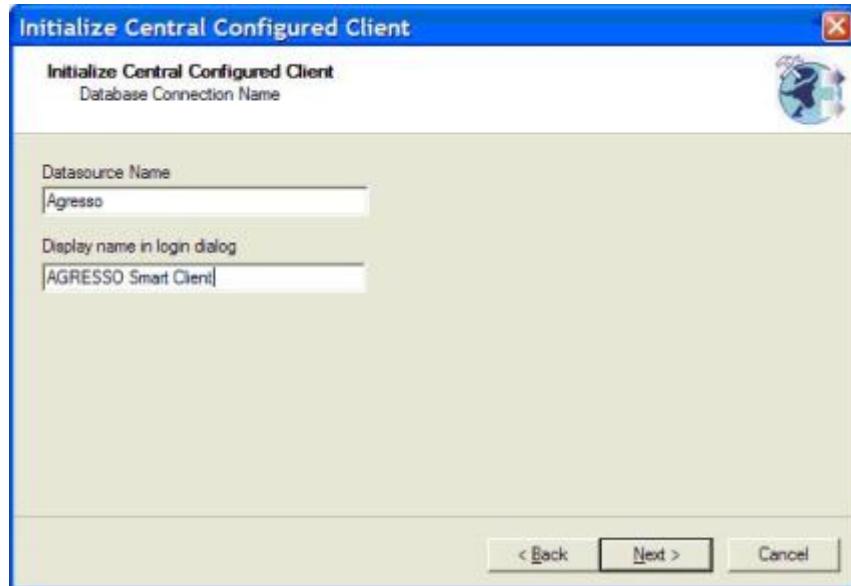
1. Select the CCC node, and then click on the [Initialize the Centrally Configured Client](#) link.

In the next dialog you will decide which version of the database files to extract from the Database CAB files folder and copied to the *bin* folders on the network share.

2. Select the database version and click **Next**.

This will open the **Database Connection Name** dialog:

DB Connection Name



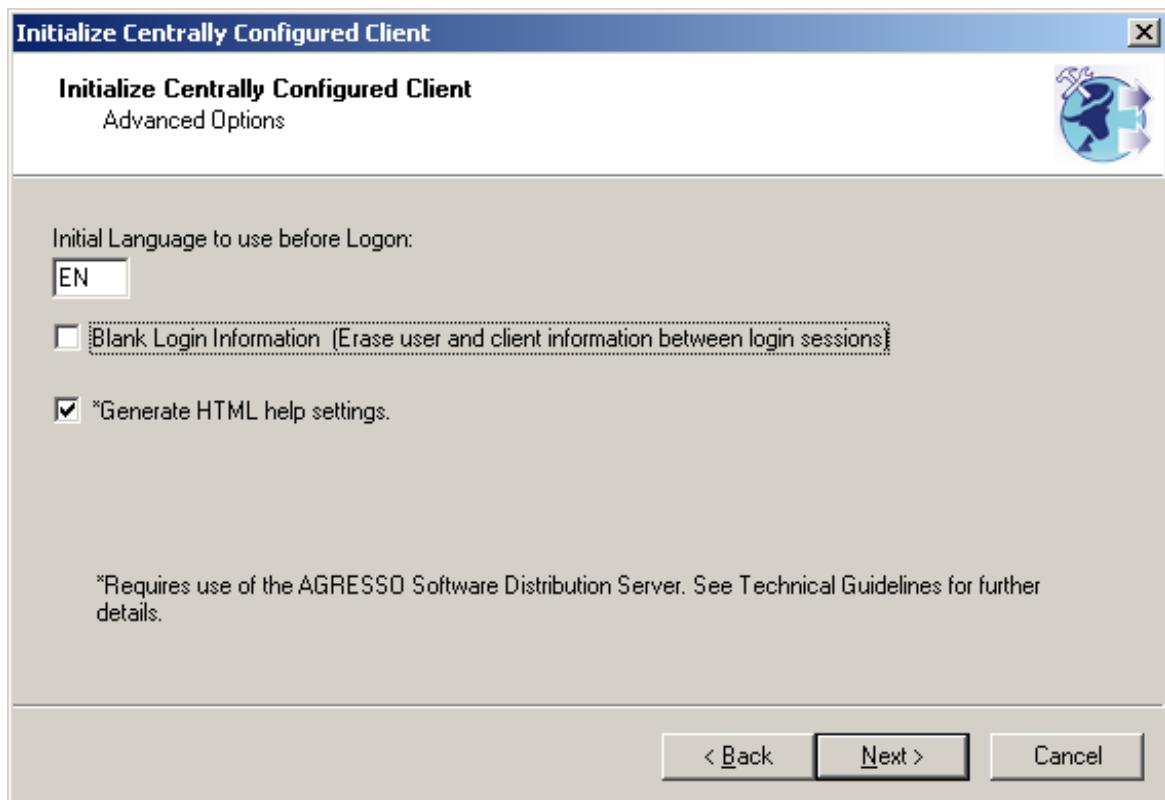
3. Enter **Datasource Name** and **Display name** and click **Next**. The **Datasource name** is the internal name used by Agresso, while the **Display name** is the name visible to the user when logging on to the Agresso Smart Client.

4. Continue according to selected database version (step 2 above):

MS SQL	Oracle
<ul style="list-style-type: none"> Enter the name of the computer hosting the database and click Next. This opens a dialog for additional connection information. Select Network Transport Protocol and click Next. 	<ul style="list-style-type: none"> Enter <ul style="list-style-type: none"> Oracle Service name Oracle SID TCP Port number Click Next <p>Note: These settings will be saved in the file tnsnames.ora.</p>

When you have entered the necessary connection information, you will be prompted to enter Advanced Options:

[Advanced Options](#)

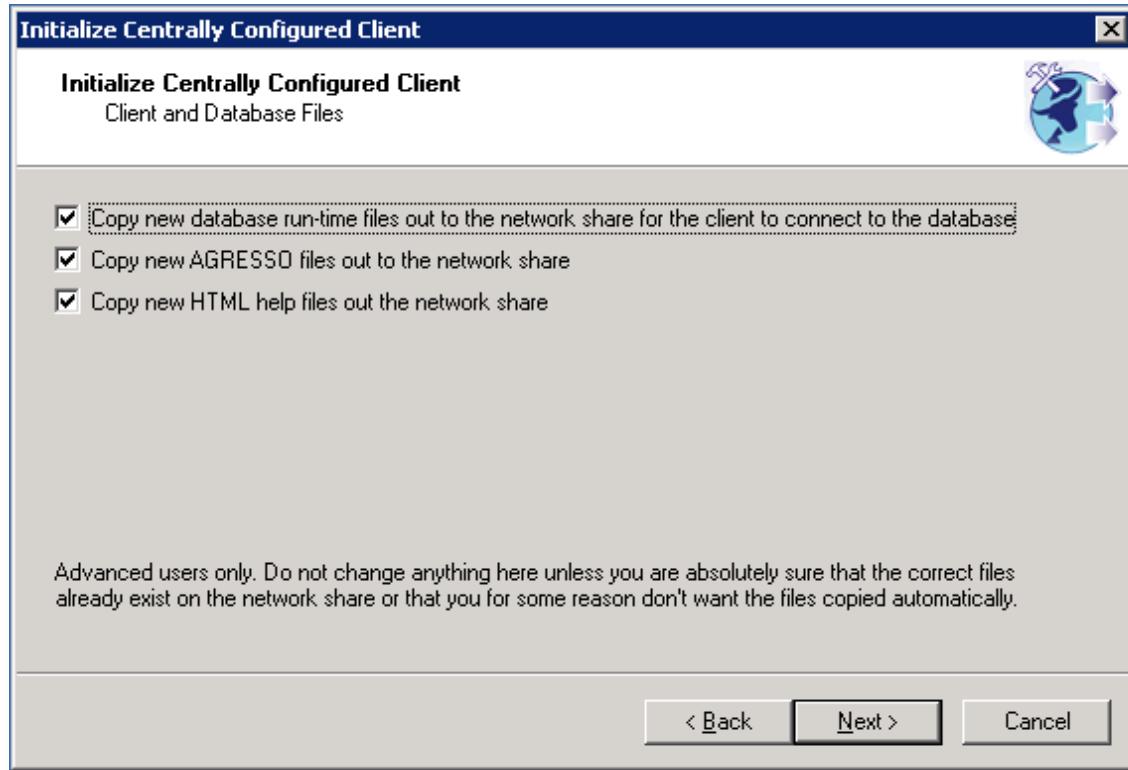


Field explanations

- If you check **Blank Login information** ... the login dialog will contain no values in the **User name** field. If not checked, the last entered user name will be displayed.
- Generate HTML help settings** is checked by default. Keep this checked if you have enabled HTML Help.
- Initial Language** ... is the language code used when displaying messages before the user is logged on to a client.

The next dialog in the wizard appears, prompting you to select files to copy:

Client and DB Files



6. Leave the default selections and click **Next** to view the files.
7. Click **Finish** to start the process of extracting and copying files to the CCC folders. This might take a few seconds.
The file *Config\Agresso32.ini* will now be created with the requested settings.

! *Authentication is configured from the smart client, see [Authentication](#)*

MAINTAINING CENTRALLY CONFIGURED CLIENTS

The CCC instance node in AMC

(Note: *instance* on the next line refer to the client you just created - located under the CCC node.)

Navigate to the **Agresso Centrally Configured Client instance** node in AMC to view and maintain the settings for the specific client. Note that any changes will be written to the *Agresso32.ini* file on the fly.

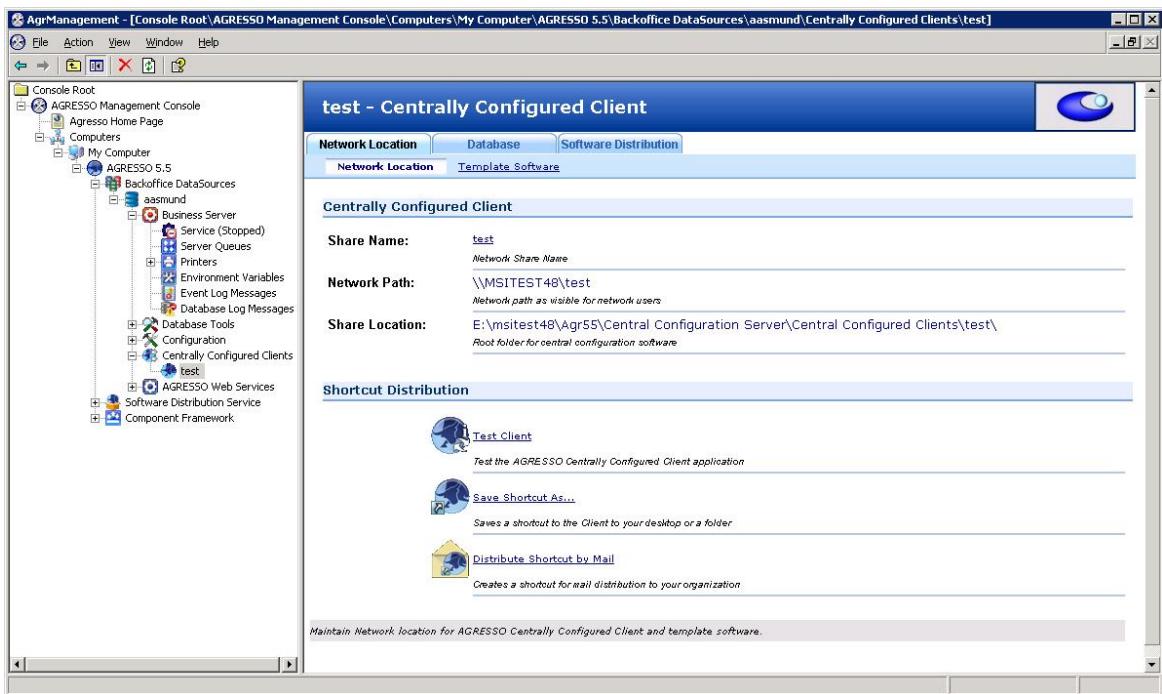
The main CCC instance screen gives you access to three tabs:

- **Network Location** - with two options: Network Location and Template software.
- **Database**.
- **Software Distribution**.

The Network Location tab

Network Location

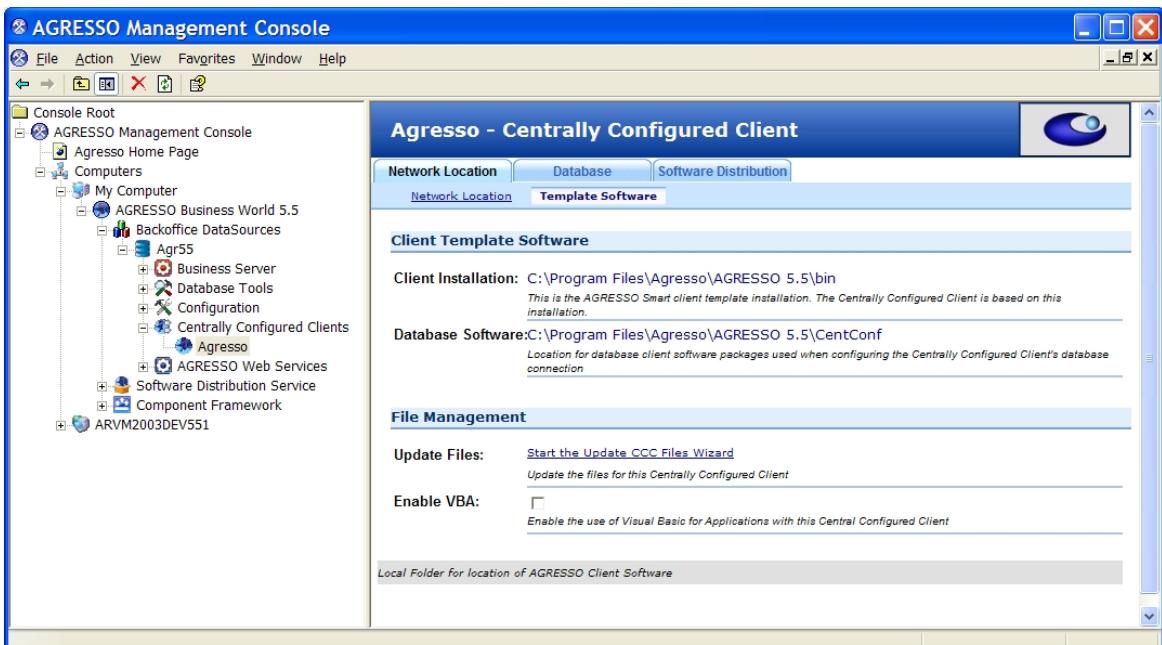
Example: CCC Network location tab/Network Location



Use the Shortcut Distribution section to create a shortcut for test and distribution to the Agresso users.

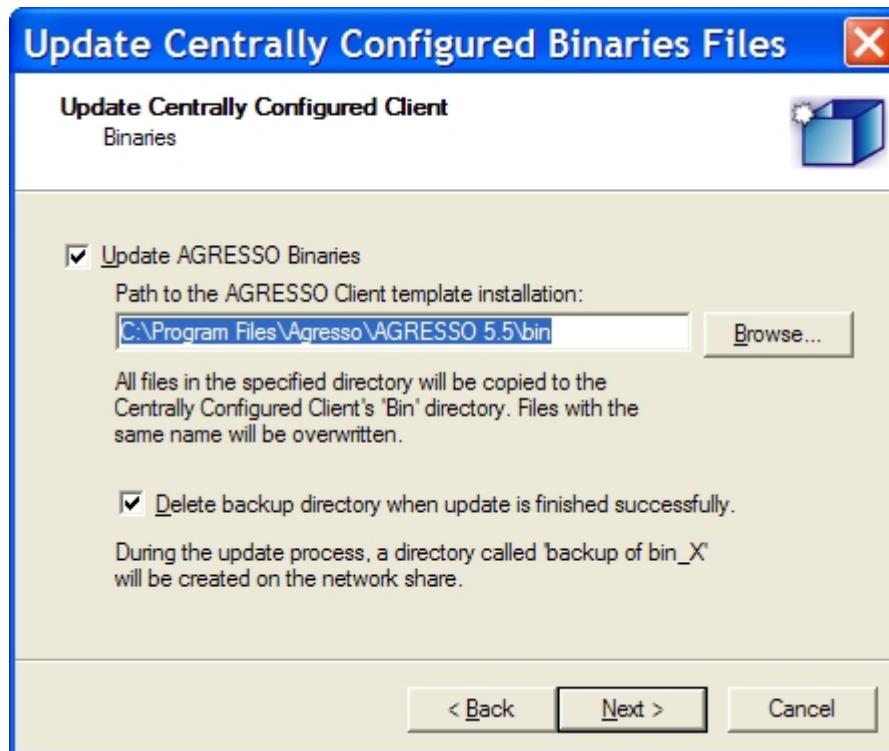
Template Software

Example: CCC Network location tab/Template Software



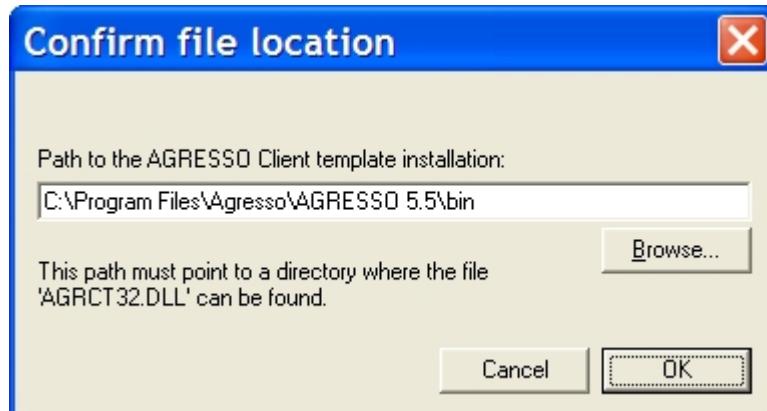
Update files: You can update Agresso Smart Client and database client files if a patch or service pack has been applied to the master installation. To update files belonging to a CCC, start the [Update CCC Files Wizard](#).

[Update Binaries](#)



The path presented will be the path to the template installation of Agresso. You can change this path if you have a custom directory where the required files are found. Before any file copying takes place, a backup directory will be made on the network share. You can choose whether or not to keep the backup files.

Path to Client Installation



By un-checking **Update Agresso Binaries**, no action will be taken on the client files. This may be useful if you just want to update the database files.

! If you install a patch or Service Pack from Agresso, only the template installation will be updated. You must run the Update CCC Files Wizard for each CCC that should have the new files.

VBA note: The files in the **CCC Bin** directory will by default be copied from the installed Agresso bin folder. This means that AgrCT32.DLL will be copied even if you do not plan to run VBA on the client computers. If these files are present in the CCC Bin directory and the client computer do not have VBA installed, the Agresso client may crash.

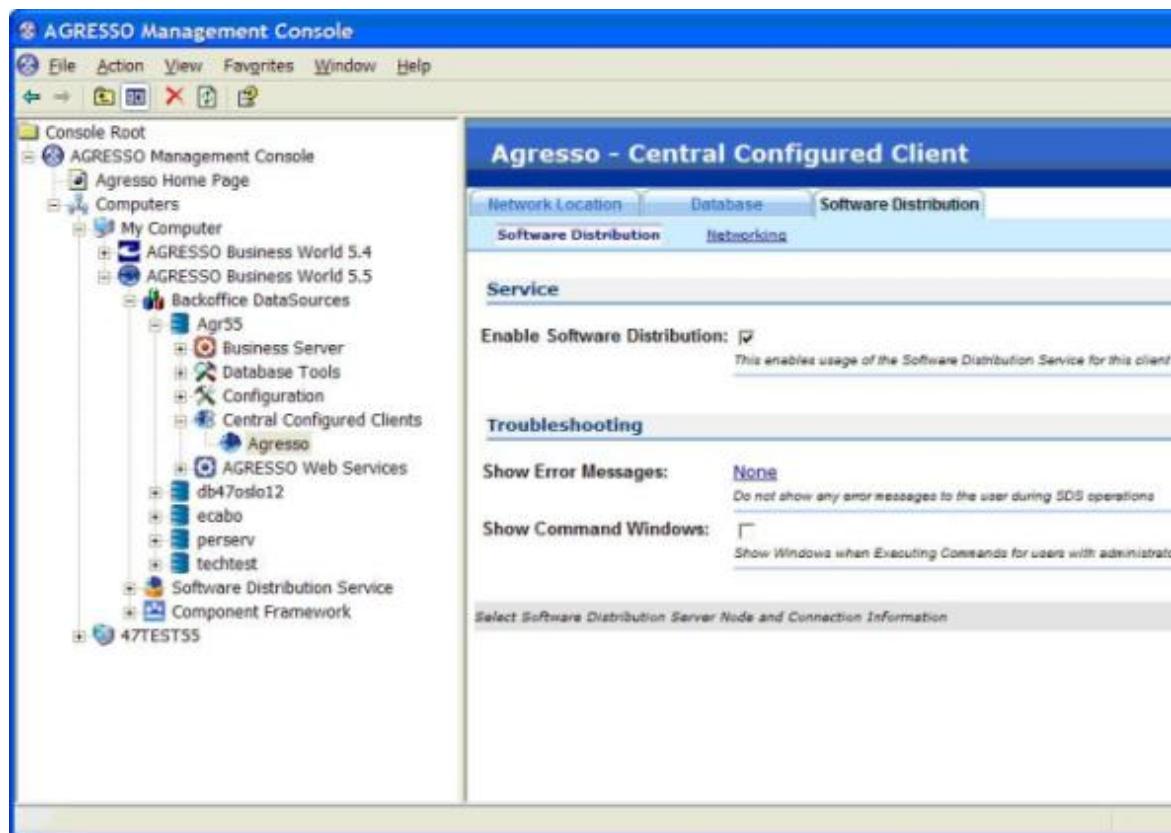
You should either install VBA on the clients or de-select the use of VBA. De-select VBA by clear the **Enable VBA** checkbox found in on the **Network Location** page under the **Template Software** context menu.

If you later want to enable VBA, check the appropriate checkbox and you will be prompted for the location of the file. The path will be the path to the template installation.

The Software Distribution tab

The **Software Distribution** tab contains the configuration needed to integrate the CCC with the **Agresso Software Distribution Server** (SDS).

 Example: CCC Software distribution tab/Software distribution



To enable SDS for a CCC, select the **Enable Software Distribution** checkbox. Also, assure that the Networking Configuration settings match SDS. See the section [Agresso Software Distribution Server](#) on how to do this.

CLIENT DATA SOURCE SETUP

Create Data source

When the Smart Client installation is completed you will need to create an Agresso Client Data Source to be able to communicate with the database.

The Agresso Client Data Source is created using the program [AgrConf.exe](#). When the Smart Client installation is completed, the last screen gives you the option to launch AgrConf directly. AgrConf is located in the bin directory. However, you will also find it as a shortcut on the **All Programs** menu; **Start | All Program | Agresso Business World Route 66 | Agresso Client Configuration**

An Agresso Client Data Source is used by the Agresso Smart Client to obtain the database connect info when starting the Smart Client.

The Agresso Data Source contains information about the RDMBS including database, database user and password. A native connection can be used for ORACLE instead of setting up an ODBC connection. If using ORACLE, the connection info is referred to the network alias defined through the Network Manager or SQL Net Easy Configuration. If MSSQL is used, you will need to create an ODBC Data Source.

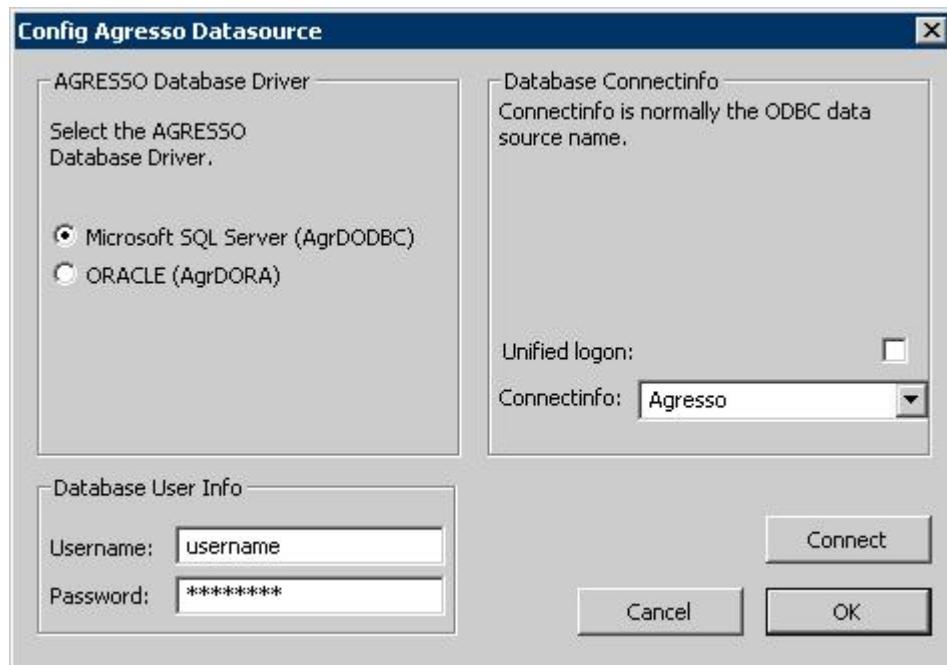
On the start up page of AgrConf you will see a list of all Agresso Client Data Sources created on the current machine. You want to create a new one. Click Add, and type a name that you want to use for the Agresso Client Data Source that you are going to create.

 Agresso Data Source



The Agresso Data Source configuration page opens. Select the Agresso Database driver that you are going to use, and type in the user name and password. If Microsoft SQL Server is used, an ODBC Data Source can be selected from the dropdown list. On Oracle you will need to type in the SQL *Net Connect name (TNSNAME or easy connect string).

Config Agresso Data Source



Click Connect to test your connection. If successful, you have now set up your Agresso Client Data Source, and you can start Agresso.exe to logon to the Agresso Smart Client using the Agresso Data Source just created.

The Agresso Client Data Sources are stored in registry under
HKEY_CURRENT_USER\Software\Agresso\<product name>\Client\Datasources.

Related topics

[Agresso on MS SQL Server - ODBC in a 64-bit Windows environment](#)

Software Distribution Server

SOFTWARE DISTRIBUTION SERVER OVERVIEW

The Software Distribution Server (SDS)

The **Software Distribution Server** (SDS) replaces the **Distributed Applets** (*agrcom.lst* and *agrinst.lst*) for locked down environments. SDS installs MSI packages, copy files to the client, run commands, and add registry settings.

This eliminates the need for the client to hold the Administrator token during startup.

SDS helps to overcome lacking access rights, when the Agresso Client software shall run from a network share. Necessary access rights are needed to create the MS SQL Server ODBC Driver entry, and when installing packages like Microsoft Visual Basic for Applications (VBA).

How SDS works

SDS is supposed to run as a domain user with administrator privileges on the client computers and will act as the administrator during software installation. When a Smart Client starts, it will send a request to SDS and ask SDS to do a check of the software installed on the client computer. SDS will then remotely deploy any missing components.

During installation, SDS will remotely control **Windows Installer** at the clients using Windows Management Instrumentation (WMI). SDS commands the **Windows Installer** at the client computers to install or uninstall MSI packages.

The administrator can configure SDS to deploy different set of packages to different users or group of users. Also, the administrator can configure SDS to remove packages, as well as upgrading packages for users or group of users.

Server/Client Authentication

SDS registers an authentication service to use for authenticating the remote procedure calls (RPC). By default LDAP is used as authentication service. The client must specify the same authentication service specified by SDS.

Major changes from 5.4

Major changes:

- Use of Null Session Shares is by default not used. Instead the MSI files will be copied from the share by the client and stored locally before SDS executes the installer.
- Server/Client Authentication
 - a. SDS registers an authentication service to use for authenticating the remote procedure calls (RPC). By default LDAP is used as authentication service.
 - b. The client must specify the same authentication service specified by SDS.
 - c. These changes were required for SDS running on XP SP2, Windows 2003 and later OS versions.

Best practice for software distribution

Agresso Software Distribution Server is not the best practice for remote administration and deployment of client computers. It is specially designed to run in a Microsoft environment with Active Directory and *does not support* Novell networks.

You should consider other tools for distribution of software and configuration settings throughout your organisation. Two options are:

- ZENworks for Novell
- SMS or Active Directory for Microsoft environments

CCC AND SDS INTEGRATION

SDS functionality

Agresso Software Distribution Server can distribute registry keys, perform file operation and execute commands at the local clients. These steps are described in an XML file named *AgrSDSConfig.xml*.

AgrSDSConfig.xml

The file *AgrSDSConfig.xml* is found in the *Config* folder on the network share. It is only used if SDS is enabled on the CCC and running at the server.

In addition to the msi packages, the file contains configuration information to be set by SDS on the client computers. Using this file you can specify 'configuration packages' – a set of operations to run in sequence during the startup of the Smart Client.

Supported operations are:

- Setting registry keys
- Copy files
- Execute Commands

The file is written in XML format and you can use macros that allow a great deal of flexibility. The following file will create the required MS SQL Server ODBC Driver entries at the clients:

```
<?xml version="1.0" encoding="utf-8" ?>
- <centralConfig>
  - <configPackage Name="Create CCC Help settings">
    - <registry>
      - <regKey Name="ItssRestrictions" Location="HKEY_LOCAL_MACHINE"
        Path="SOFTWARE\Microsoft\HTMLHelp\1.x" appendIfMissing="true" delimiter="/">
        <regValue Name="UrlAllowList" Type="string">$AGR_HELP$\Default</regValue>
        <regValue Name="UrlAllowList" Type="string">$AGR_HELP$\EN</regValue>
        <regValue Name="UrlAllowList" Type="string">file://</regValue>
      </regKey>
    </registry>
  </configPackage>
</centralConfig>
```

Macros

The following macros can be used when creating Configuration Packages in the *AgrSDSConfig.xml* file:

Folders:

```
$ADMINTOOLS$  
$COMMON_ADMINTOOLS$  
$APPDATA$  
$COMMON_APPDATA$  
$COMMON_DOCUMENTS$  
$COOKIES$  
$HISTORY$  
$INTERNET_CACHE$  
$LOCAL_APPDATA$  
$MYPICTURES$  
$PERSONAL$  
$PROGRAMFILES$  
$PROGRAM_FILES_COMMON$  
$SYSTEM32$  
$WINDIRS$  
$MYMUSIC$  
$RESOURCES$  
$RESOURCES_LOCALIZED$
```

Other:

```
$COMPUTERNAME$ - Name of the client Computer  
$USERNAME$ - Name of user  
$USER_SID$ - User Account SID  
$AGR_BIN$ - CCC Bin folder path
```

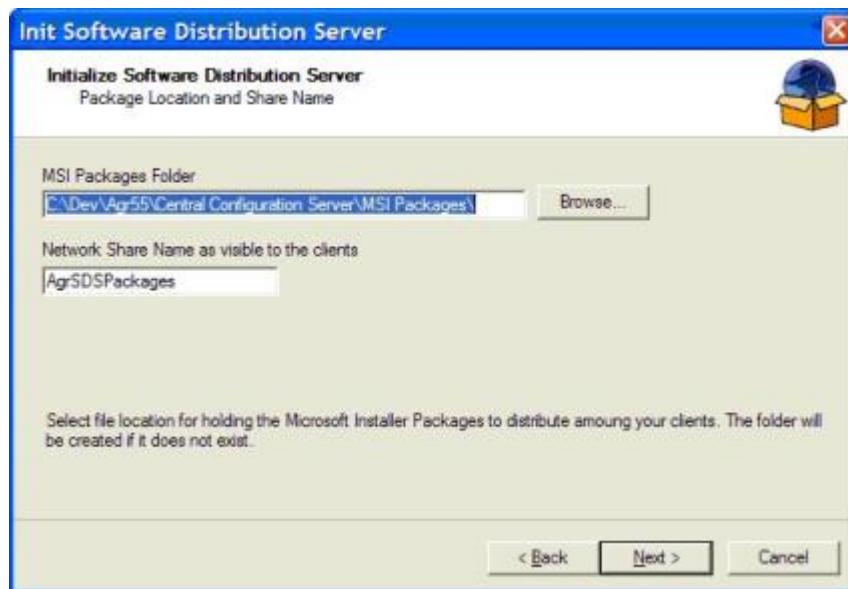
INSTALL THE SOFTWARE DISTRIBUTION SERVER

Installation procedure

To install the Software Distribution Server - open the **Software Distribution Server** node in AMC and select **Initialize the Software Distribution Service**. This will start a wizard to guide you through the process.

1. Click **Next** in the **Welcome** window, to open the **Package Location and Share Name** dialog:

 Example: Package Location and Share Name



2. Do as follows:

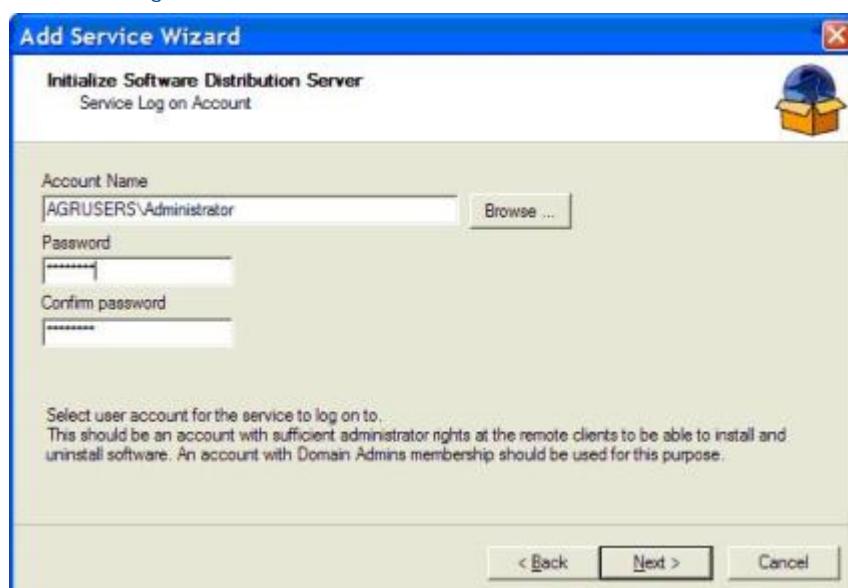
- Assure that the MSI Package Folder is correct.
- Enter a name for the Network Share. Default value is **AgrSDSPackages**

The folder and the network share will be created with the following properties set:

- Folder
Full Control to the user that created the folder (Administrator)
Read & Execute, List folder content, and Read for Everyone
- Share
Full Control to the user that created the share (Administrator)
Read access to Everyone
- Click **Next**.

This will open the **Server Log On Account** dialog.

 Service Log On Account

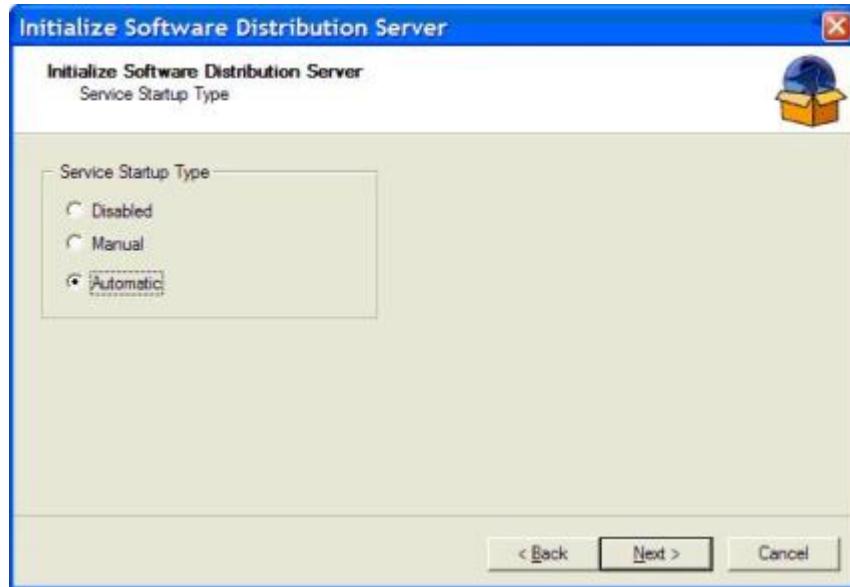


Please note: Assure that the User Account has local administrator privileges at the server computer or sufficient privileges to write to **HKEY_LOCAL_MACHINE\Software**.

! It is crucial that the User Account privileges are sufficient to install and uninstall MSI Packages at all client computers accessed by users of Agresso.

3. Assign a User Account with the necessarily privileges and enter password. Then click **Next**. This will open the **Service Startup Type** dialog.

 **Service Startup Type**



4. Set start-up type to **Automatic** (recommended) and click **Next**.

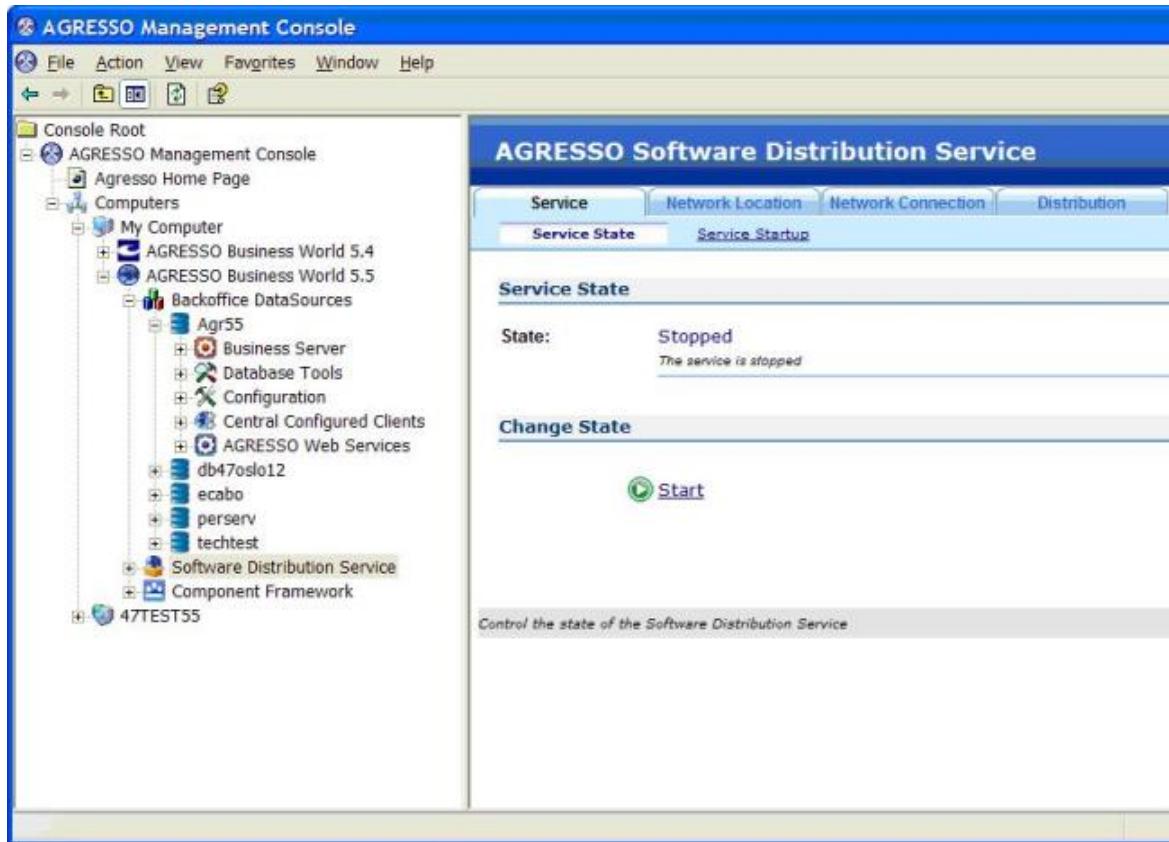
5. Click **Finish** to complete the installation.

MANAGING THE SOFTWARE DISTRIBUTION SERVER

General

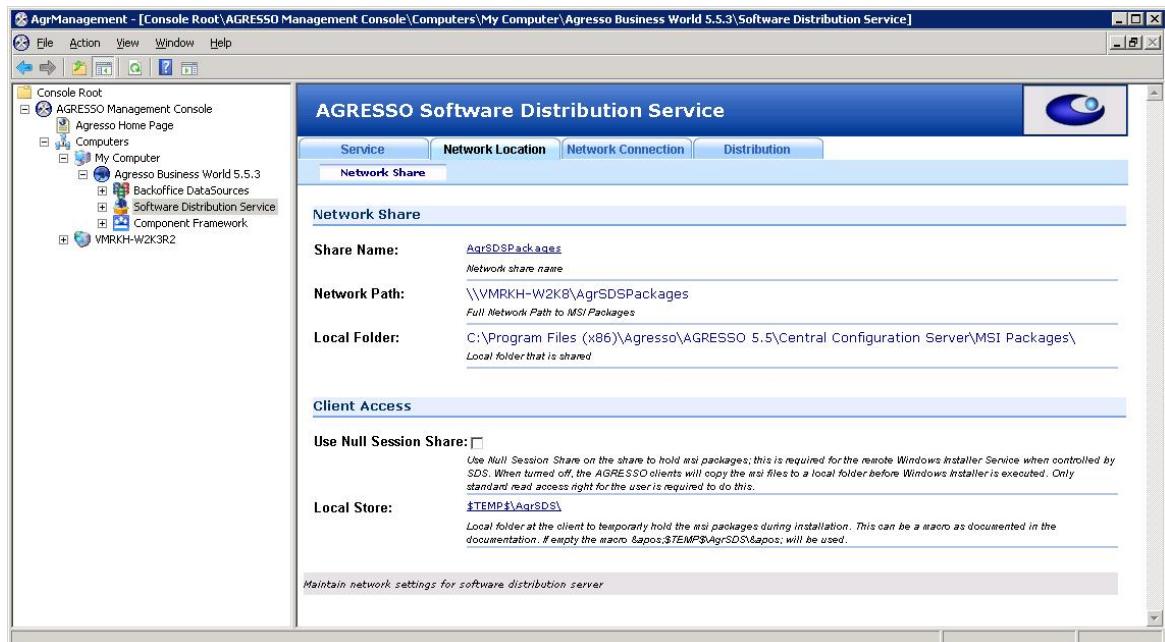
The **Agresso Software Distribution Service** page is used to manage the service. You can control the service state as well as changing startup properties like log on account and password.

 **Example: Software Distribution Server Page**



The **Network Location** tab displays the network share name and the local folder location for MSI Package distribution.

Example: Network Location



You can set the following options under Client Access:

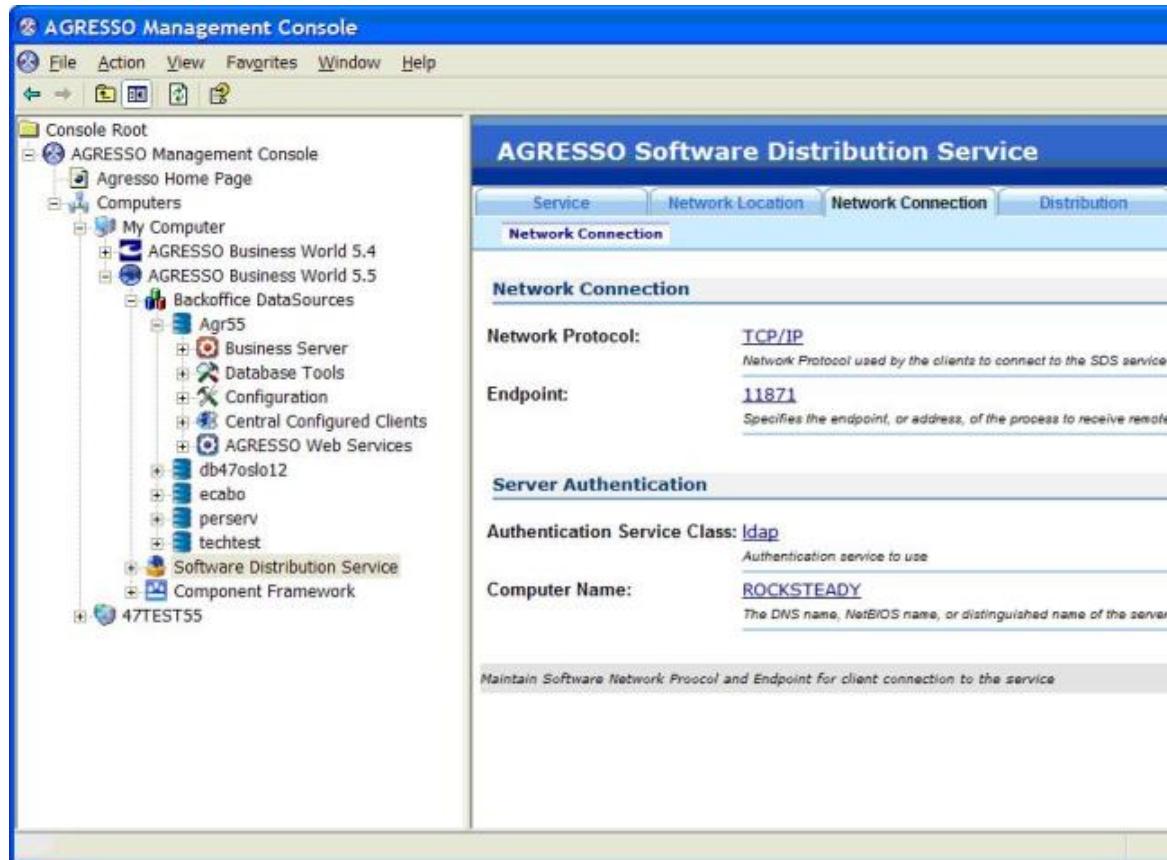
- **Use Null Session Share**

Turn on use of Null Session Share for the share holding MSI packages. This is used to allow the remote Windows Installer Service to access the network share. When turned off, the Agresso Smart Clients will copy the MSI files to a local folder before Windows Installer is executed. In this case only standard read access right is required (recommended).

When turned on, the share name will be added to the "NullSessionShare" list to be accessible to services as Windows Installer (the remote Windows Installer will be running in the context of the WMI Service.) Null Session Shares are considered to be a security risk.

- **Local Store**

Local folder at the client temporarily holding the MSI packages during installation. This can be a macro. If empty, the macro '\$TEMP\$\AgrSDS\' will be used. This will instruct the Smart Client to copy the MSI package off the share and to a folder named AgrSDS located below the user's temp folder, e.g. "C:\Documents and Settings\user.name\Local Settings\Temp\AgrSDS".

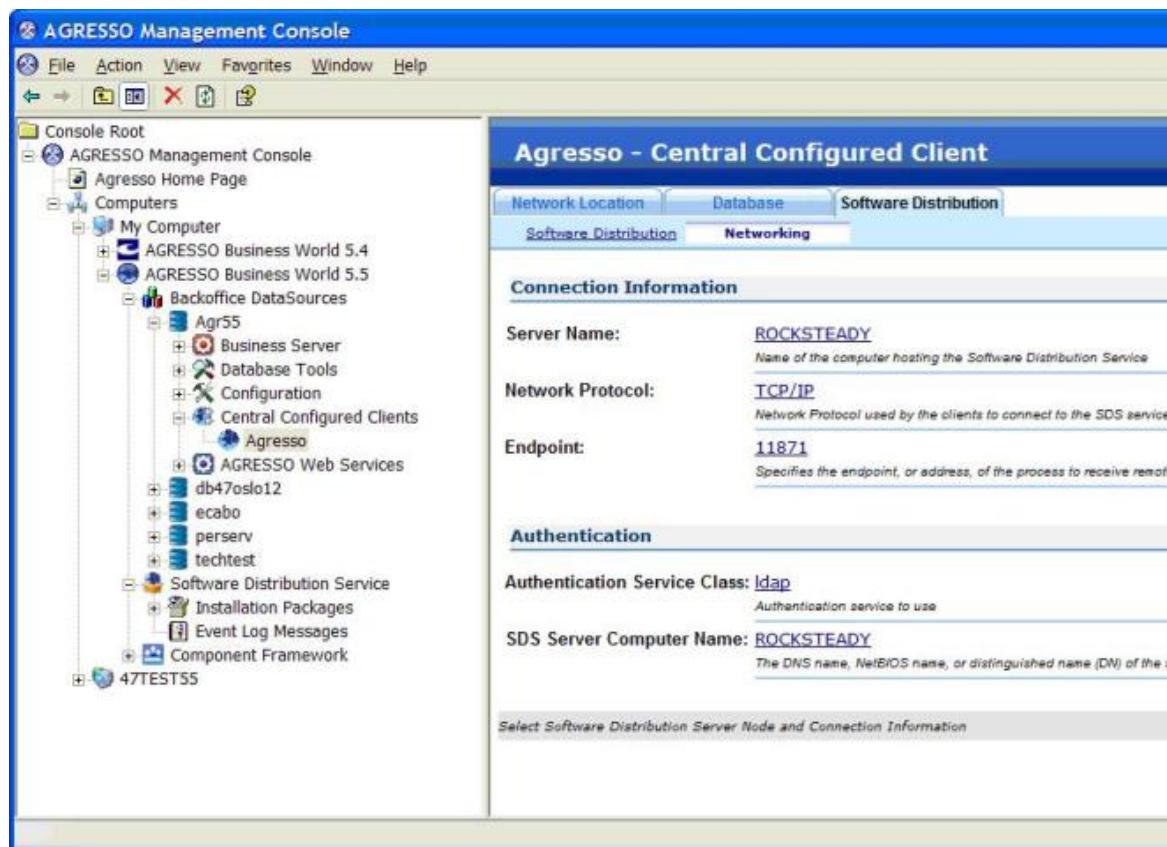
 Example: Network Connection


Use the **Network Connection** tab to set network protocol and end point for the service, used when receiving networking requests from Agresso Client.

You can choose between **TCP/IP** and **NamedPipes** as network protocols.

 *This information must correspond to the networking information set in the Central Configured Client. If this does not match the client will not be able to connect to the SDS service.*

 Central Configured Client - Networking Information



ADDING INSTALLATION PACKAGES

Adding Installation Packages

The purpose of this task is to locate the MSI files and to copy them to the shared folder. You find the MSI packages installed with Agresso in the *DistApp* directory.

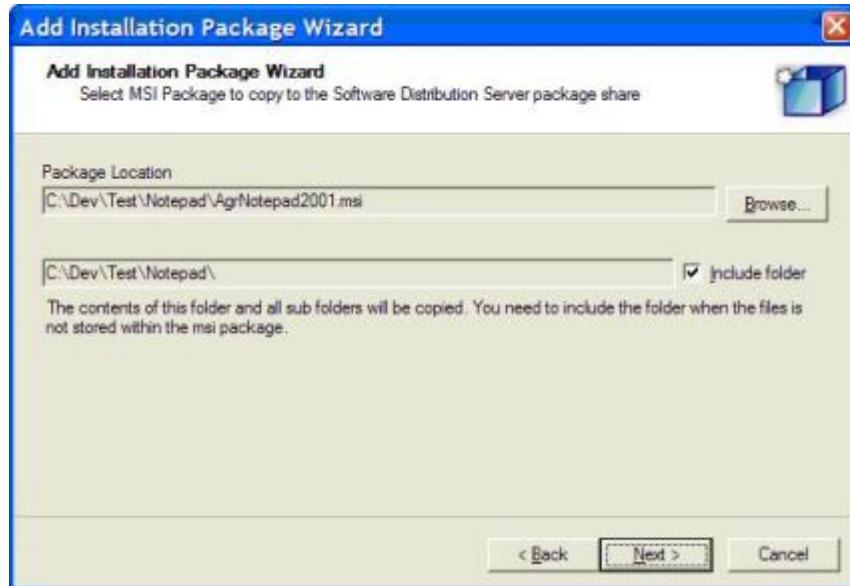
There are two types of MSI files:

- MSI files that contain both installation information and all files within the MSI package itself (one single file).
- MSI files that contain installation information, but all files are outside the MSI package and might consist of subfolders as well. In the last case mark the Include folder check box.

Do as follows:

1. Right click the SDS node and select **New | Add Installation** to start up the **Add Installation Package Wizard**.





2. Specify the location and click **Next**.

The next step is to specify additional parameters required for the specific MSI file during installation. The parameters are appended to the standard parameters when starting windows installer.

3. Add parameters (if needed) and click **Next**.

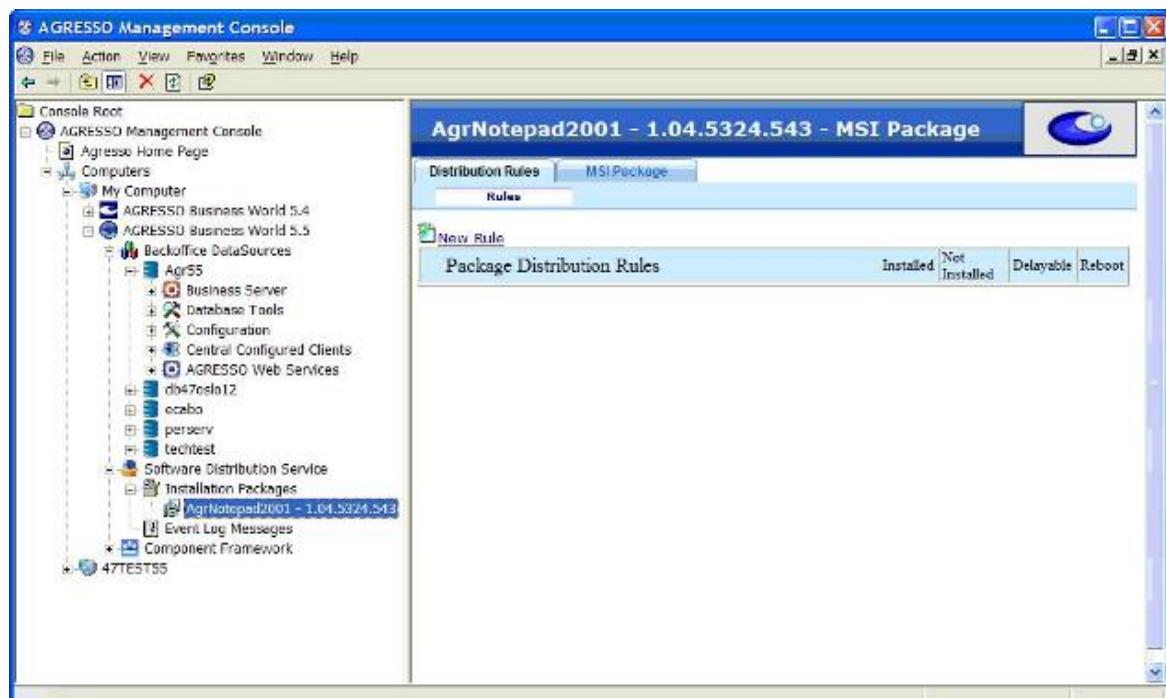
You can close the wizard.

Distribution rules

Adding an MSI Package to the distribution share will not in itself cause the SDS service to push the package. The Distribution Rules are set on each package to tell which users that should have the package installed or uninstalled. No action will be performed for users not listed in any of the rules.

You should at least create one distribution rule per MSI Package to control how the package should be distributed in your organization.

Package Distribution Rules



Adding a Distribution Rule

You can assign rules to users directly by adding the user's domain name, or indirectly by adding a Domain Group where the user is a member.

When a user's domain name is added to a rule, that rule will take precedence over conflicting rules. If you want to ensure that a specific rule is used for a specific user, add the user account to the rule.

Add a new rule: Do as follows:

1. To add a new rule:

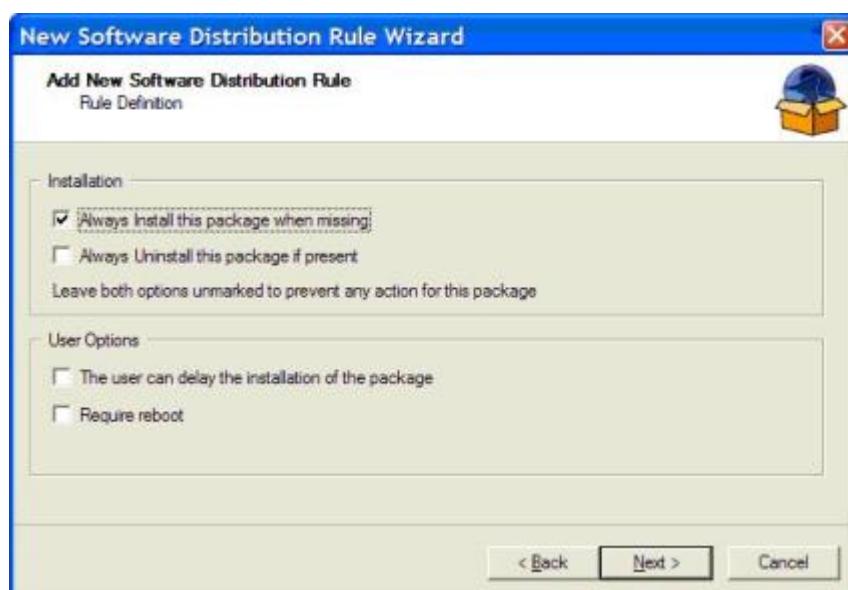
- a. select the **MSI package** node and select the [New Rule](#) link found on the top of the **Distribution Rules** tab. This will open the **New Software Distribution Rule Wizard**.
- b. Click **Next** to continue.

Example: Software Distribution Rule Name and Description



2. Specify the rule name, enter a rule description and click **Next**.

Rule Definition



In the Rule Definition window you can create three types of rules:

- Install rules with the install flag set
- Uninstall rules with the uninstall flag set
- No-Action rules with none of the two flags set

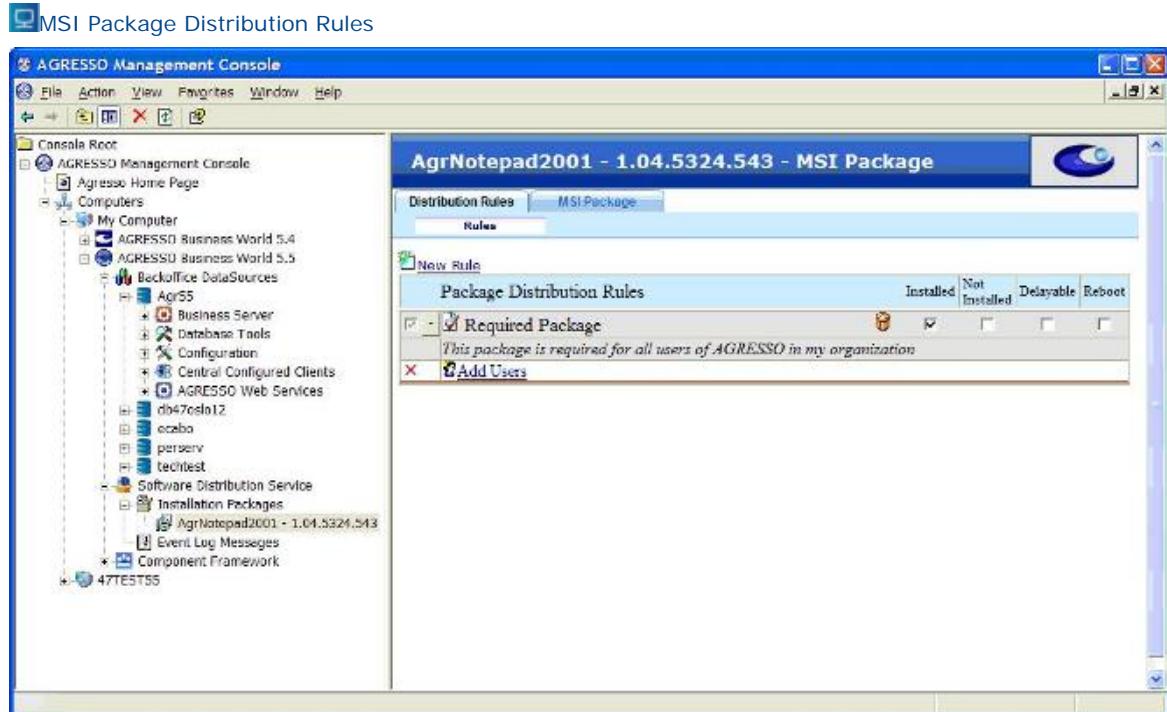
Moreover, you can specify if the Rule Members should be able to delay the package installation until next time the user starts Agresso Client. You can mark the rule to require a reboot after install/uninstall of the package.

3. Make your selections and click **Next**.

You have now added a rule.

Maintain Distribution Rules

Use the **MSI Package Distribution Rule** page to maintain the rules by adding users or groups to it, or to remove or add additional rules. Here we give a general overview of the tasks involved.



1. Select the Add Users link located below the rule definition. This will bring up the standard **Select Users or Groups** dialog.

Select Users or Groups



2. Add users and groups and click **OK**.

The user's or group's Domain Names are added to the package rule.

Delete: To delete a group or user, select the leftmost check box and use the X-button. If you would like to select all users and groups listed in a specific rule, select the check box at the top of the rule.

Rules Priority

A rule is followed if:

- The user is a direct member of the rule
- The user is an indirect member of the rule and not an indirect member of any other rule

If the user is an indirect member of more than one rule, the first **Install** rule found is used, then the first **No Action** type of rule, and finally if not indirect member of those, the first **Uninstall** rule is used.

Conflicting Rules

Conflicting rules exist when a user account is added to more than one rule. The Rule manager will try to permit this from happening.

This document discusses how to add and upgrade the packages to be distributed. This is done from the Installation Packages node.

How to upgrade Installation Packages

If you apply an Agresso Service Pack with new versions of MSI packages, you should note that only the source packages will be updated. You must run the **Add Installation Package Wizard** for each new package and also set up the rules (see above) before the package is pushed out to the users.

You can have multiple versions of the packages, even if they have the same name. This means that you do not have to delete the old ones before you add new versions. You can just update the distribution rule.

The Agresso Self Service client

ADDING THE AGRESSO SELF SERVICE CLIENT

Prerequisites

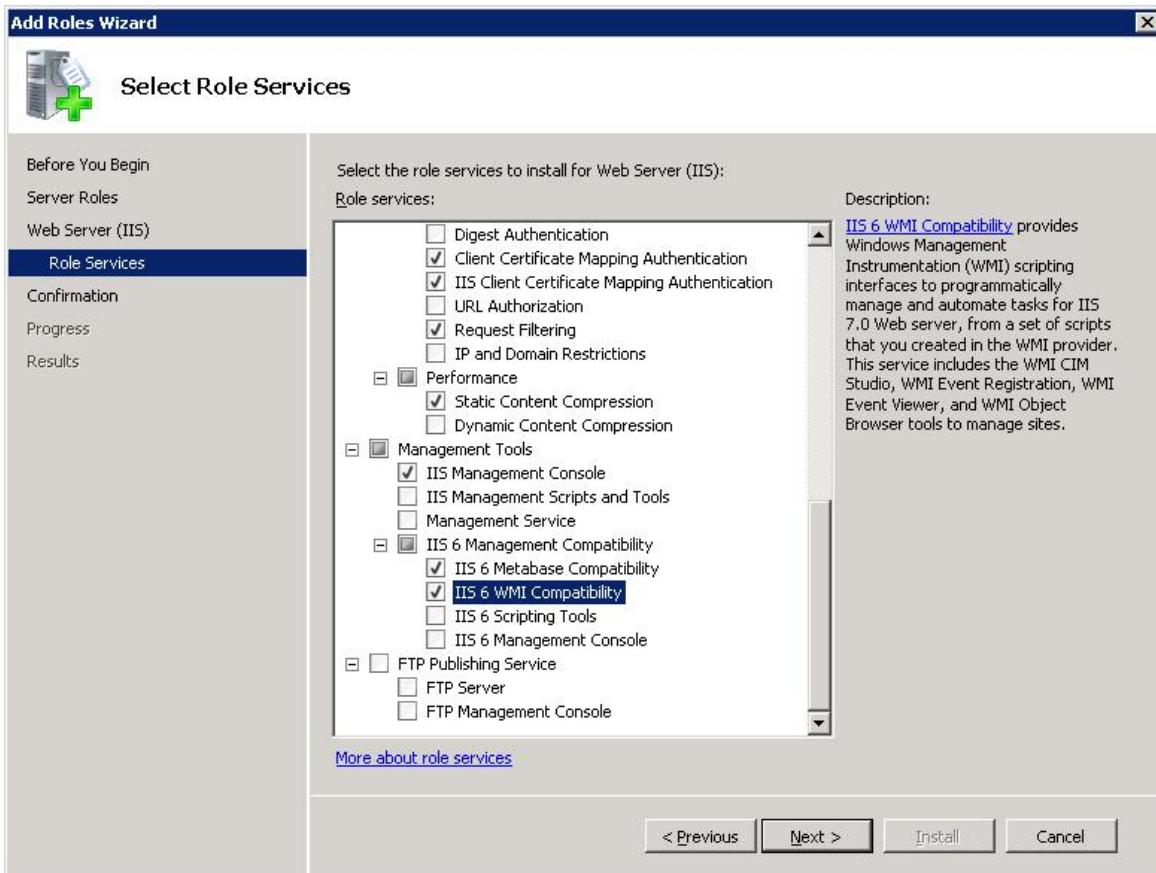
You use the **Agresso Management Console** (AMC) to set up and configure the Agresso Self Service client. In order to do that, however, **Internet Information Services** (IIS) must be installed on the computer, and **Agresso Web Services** must have been selected during installation.

IIS version dependencies

IIS 7

IIS 6 compatibility must be enabled to be able to configure Agresso web applications on IIS 7 from AMC. The IIS 6 compatibility components can be added using the **Turn windows features on or off** option under **Programs and features**, located in the **Control panel**. Make sure the **IIS Metabase Compatibility** and **IIS 6 WMI Compatibility** feature is selected. The Agresso Business World installation enables you to install the missing IIS features detected.

 Adding IIS6 compatibility components



Activating and de-activating Self Service

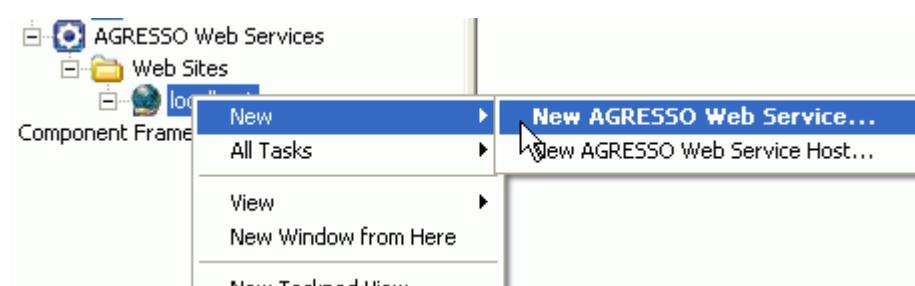
The relevant Self Service files will have been copied to the correct folders during installation. To make it available for users, it must be manually added to the AMC tree, and configured from there.

Published or withdrawn: When the Self Service client is added, it is also automatically made available for Agresso; it is *published*, and the application's *connection point* is available for public use. Once added, you find the Self Service client among the Web service nodes in AMC, under the **Web Site** node. From here, you get access to all settings, and can easily withdraw it (and later re-publish it) when necessary.

Adding Self Service

Do as follows to add the the Self Service client to the AMC tree:

1. Log on to the correct data source.
2. Right click on a **Web Site** node and select New Agresso Web Service



A Wizard will guide you through the remaining steps.

3. Follow the instructions on screen.

When finished, the Self Service client will be listed in AMC.

4. Click on a Self Service node to open the management page and continue with [Configuring Agresso Self Service](#).

Enable windows authentication

By default, users will get access to Agresso Web services through user name and password. If you want to enable Windows authentication, you must do as follows:

1. Select [Windows Authentication](#) as the preferred authenticator for the Agresso installation. You use the [Authentication Setup](#) window in ABW Smart client for this purpose.
2. Click the [Configure windows authentication](#) link (the **Authentication** tab) in the properties window for the web application (see [Self service example](#) below).

Note: This requires that the Agresso users are linked to a domain user. This is configured in [User Master File](#) (ABW Smart client).



See [Authentication](#) for more details on the authentication setup.

CONFIGURING AGRESSO SELF SERVICE

Communication protocol and security

When you configure the Self Service client, you must select among the following protocol options:

- a. [http with https login](#) – ensures secure login (password encryption), but all other communication will be http based.
- b. [http](#) – all communication, login included, is http based.
- c. [https](#) – all communication, login included, is https based, meaning that all communication is protected.

Alternatives a. and c., where https is involved, requires special attention.

HTTPS and certificates

Existing valid certificate

Communication using https requires that a valid certificate is installed on the server. If you already have a valid certificate from a trusted certificate authority, this can be used by Agresso.

Self-signed certificate

If you don't already have a valid certificate, Agresso can automatically generate a self-signed certificate through the installation process. Unless the clients are set up to trust this certificate, however, they will receive a security warning each time they try to log on to Agresso.

One of the following alternatives will solve the problem:

- a. You can replace the self-signed certificate with a valid, trusted certificate from a trusted certificate authority (recommended!).
- b. You can add the certificate to the Trusted Certificate Database for all clients involved. (Still, it is recommended to replace it as soon as possible.)

MS SQL Server notice!

Installing SQL Server on a machine with certificates may fail. If the SQL Server finds certificates on the local computer, it will attempt to use the certificates.

If the certificate is not issued to the *fully qualified domain name* of the computer, SQL Server considers the certificate as invalid. If SQL Server is installed and a certificate that is not issued to the fully qualified domain name is added, local connections will fail.

For more information, see: <http://support.microsoft.com/default.aspx?scid=kb;en-us;309398>

Document Archive

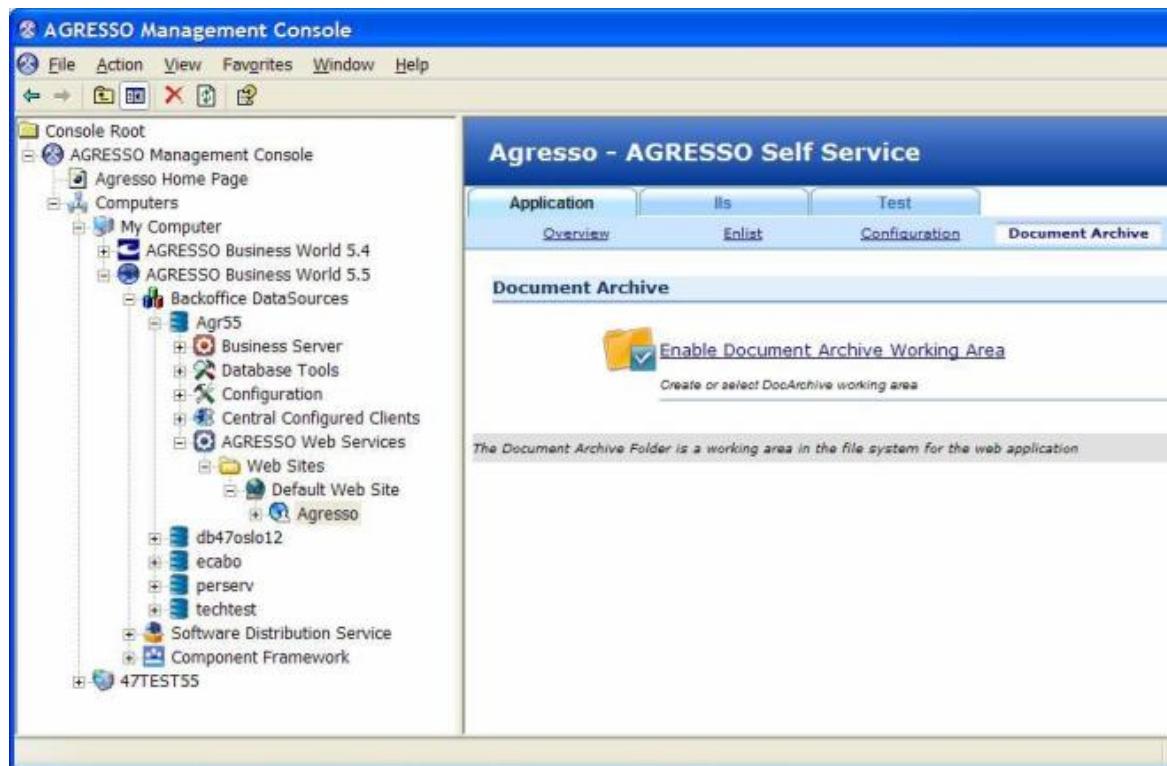
To enable use of Document Archive, a separate page is available in AMC at the **Agresso Self Service** node.

In order to enable Document Archive the following must be done:

- Select a folder to used by Document Archive as working area
- Give the anonymous access account permission to read and write documents to the folder
- Add to web.config an attribute named AgrDocTempFiles with the folder path as value

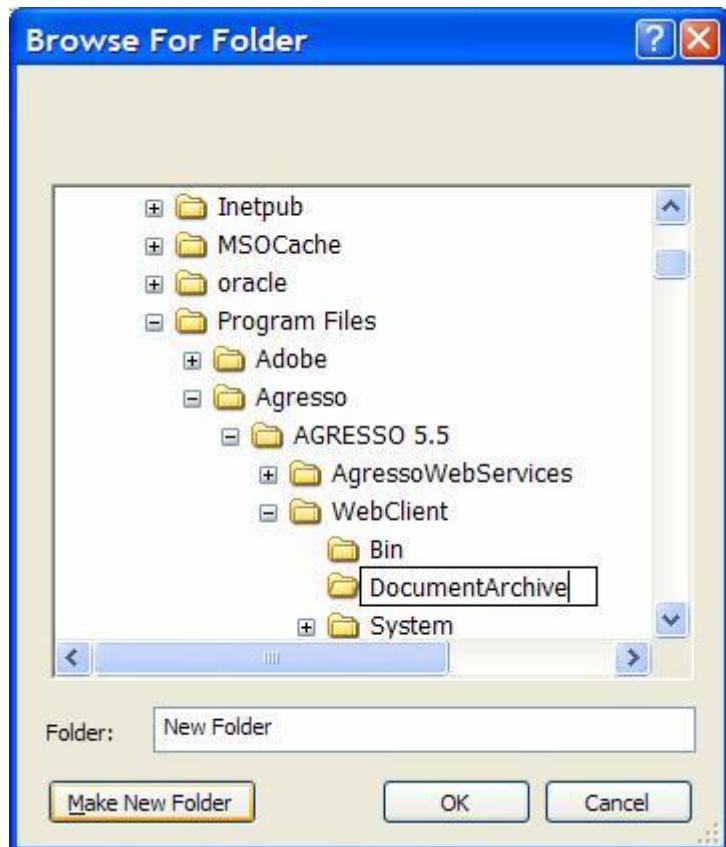
1. On the **Application** tab, select the **Document Archive** and then the [Enable Document Archive Working Area](#) link to select document archive folder.

Enabling Document Archive Area



2. Next, create a folder in a preferred location or select an existing folder.

Select Folder

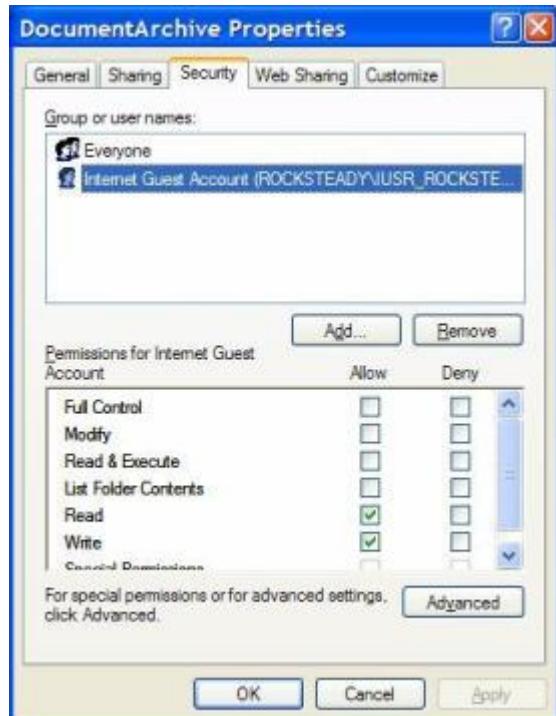


3. Once the folder is created, the permission attributes are changed to allow the Anonymous Access Account read and write access.

Anonymous Access Account

4. Browse to the folder and select **Properties | Security** if you want to verify that the permissions are correctly set.

Security Properties



5. The [web.config](#) file is changed to allow the Agresso Self Service components to look up the folder location when reading and writing files to the folder. The config file contains the following information:

```
<add key="AgrDocTempFiles" value="C:\Program Files\Agresso\AGRESSO 5.5\WebClient\DocumentArchive"/>
```

To disable the Document Archive working area select the [Disable Document Archive Working Area](#) link. This will reverse the changes except for the permission changes to the physical directory. If the AgrDocTempFiles setting in web.config is not set, the windows temporary folder will be used.

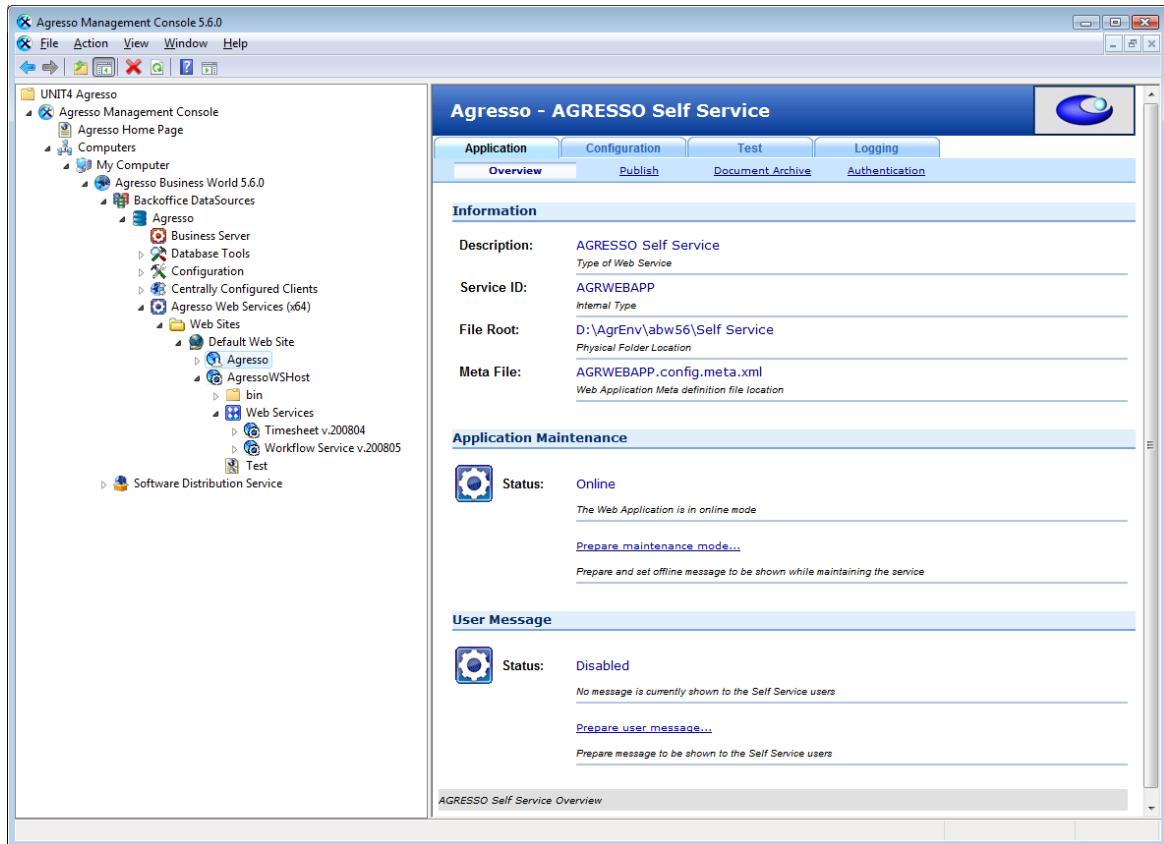
MANAGING AGRESSO SELF SERVICE

Options

You use AMC to inspect and configure the Self Service client. AMC lets you view basic information, view event log messages, change [web.config](#) attribute values, and change document archive working area (see [Document Archive](#)).

To delete a Web Service, right-click the Web Service node and click "Delete". This will remove the virtual directory and withdraw the service. This will not delete [web.config](#) or other files from the disk.

[Agresso Self Service](#)



Application maintenance

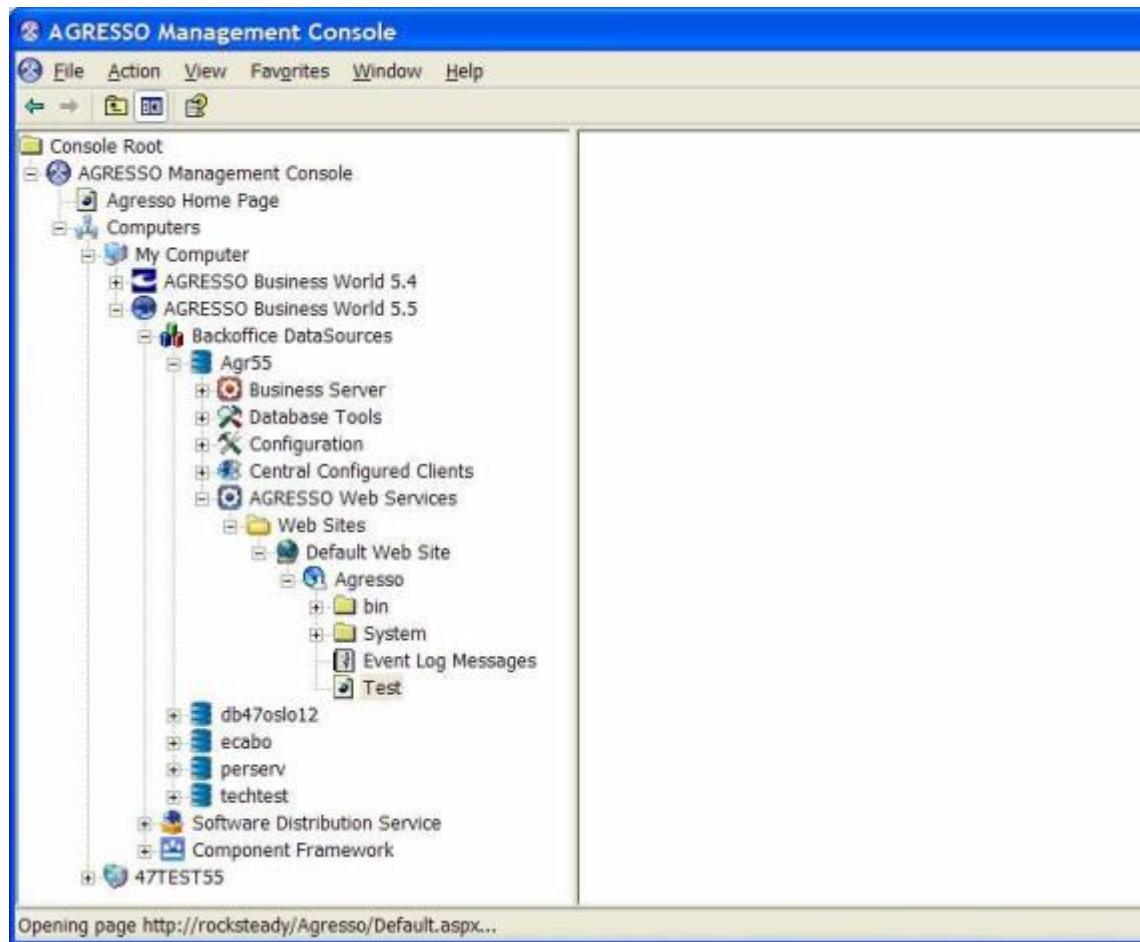
When maintenance is required, you can set the application in maintenance mode. Just click on [Prepare maintenance mode...](#) to get access to the various settings:

Application Maintenance	
	Status: Online. Set maintenance message: Set message to displayed while the application is down for maintenance
Message Title:	Agresso Business World - Maintenance
Offline Message:	Agresso Business World is offline for maintenance. We apologize for any inconvenience. Please try again later.
<input type="button" value="Set Offline..."/> Set the Web Application into maintenance mode	
<input type="button" value="Cancel"/> Cancel and return to normal view	

The Test Node

You can test the Service URI from within AMC or in an external browser. The **Web Service** node can be selected to check that the application runs properly.

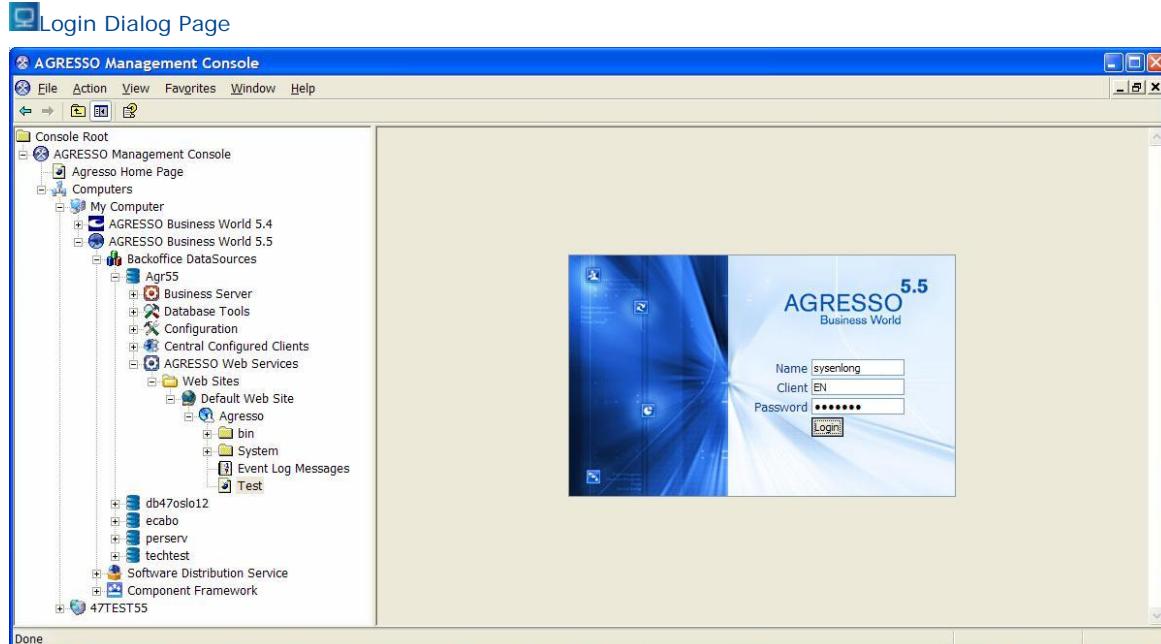




The first time a web service is accessed after installation or after a change to one of its configuration settings (e.g. [web.config](#)) has been performed, the ASP.NET layer will do a pre-compilation of the Agresso .NET assemblies and perform caching of the assemblies. This might take a few minutes. The assemblies are cached in the temporary ASP.NET files directory, located in the .NET Framework directory:

<C:\Windows\Microsoft.NET\Framework64\v4.0.30319\Temporary ASP.NET Files\<web application name>>.

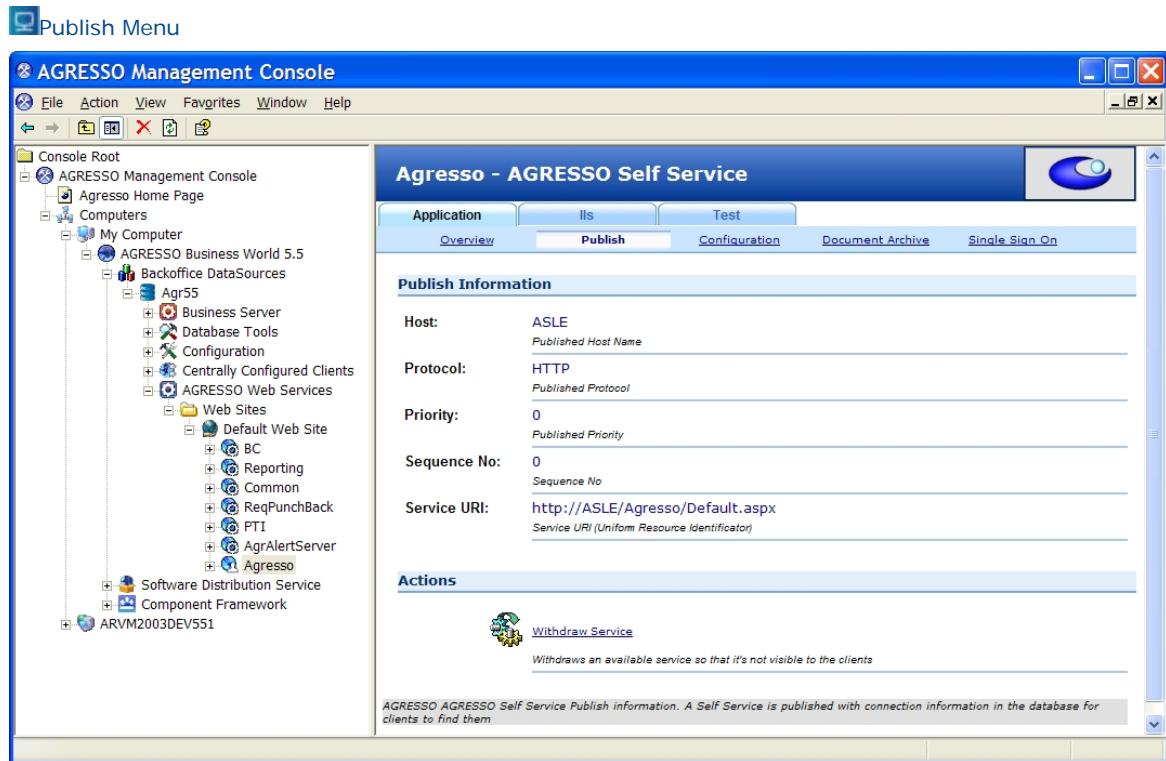
If the application is working, the log-in dialog page should appear and you should be able to log in.



Publishing and Withdrawing a Web Service

Publish

You must publish the service to make it discoverable by the client. This means to publish its lookup information in the database so that clients can request its services.



Withdraw(but not remove)

Use the withdraw function if you want to hide the web service, for instance during test or upgrade. To withdraw the service, you must:

- Select **Publish** on the **Application** tab, and then click the [Withdraw Service](#) link.
- Confirm the action.

This will change the web service entry status in *aagserviceuri* to **C**.

Make the Web service unavailable

When you retire a Web service, it will still be available on the Web server. If you want to make the Web service unavailable for external users, you should use IIS to stop it!

NETWORK LOAD BALANCING

Introduction

When the number of users of the Self Service Client exceeds the capacity of one Web server, you will need to add more servers, and connect them in a cluster.

Network Load Balancing (NLB) is a clustering technology offered by Microsoft.

NLB uses a distributed algorithm to load balance network traffic across a number of hosts (Web servers), helping to enhance the scalability and availability of mission critical services. It also provides high availability by detecting host failures and automatically redistributing traffic to operational hosts.

Steps to activate NLB

Overview

You will use an appropriate network configuration program, to activate and configure the NLB cluster.

To activate NLB in your environment, you must do the following:

1. Connect the servers with cables and cluster software.
2. Assign a unique IP-address to each server, and a common IP-address for the cluster (i.e. for all servers).
3. Set up basic rules for how the cluster shall handle user requests.
Note: If the users are behind a fire wall, you must select Affinity = **none**. This will involve some overhead, and may lead to performance problems. Otherwise, use Affinity = **single**.

Reference

Please refer to [Microsoft TechNet](#) for further details about NLB.

Agresso Web Services

ADDING AGRESSO WEB SERVICES

Prerequisites

You use the **Agresso Management Console** (AMC) to set up and configure the Agresso Web services. In order to do that, however, **Internet Information Services** (IIS) must be installed on the computer, and **Agresso Web Services** must have been selected during installation.

IIS requirements

See [IIS version dependencies](#).

External and internal Web services

Agresso has developed a set of Web services for general use by our customers. These are essentially new tools for customisation, allowing developers to create specific Agresso applications - in addition to the possibilities found in ACT.

Opposed to these external Web services, we have a few internal ones, designed and developed to be used by other Agresso components.

Activating a Web service

The relevant (Web service) files will have been copied to the correct folders during installation. In other words, all the Web services are available, but not yet configured and ready to use.

Added: To make a Web service available in the AMC tree, it must be manually added by using the **Agresso Management Console**.

Published or withdrawn: When a new Web service is added, it is also automatically made available for Agresso; it is *published*, and the service's *connection point* is available for public use. Once added, you find the Web service nodes -and/or the Web Service Host node in AMC under the **Web Site** node from where you can *withdraw* it (remove the connection point). Even as withdrawn, it will still be visible under the Web Site node in AMC, and you can easily publish it again.

Available Web services

Internal services

Agresso comes with the following internal Web services:

Internal Web service	Internal name	Description
Report engine	REPORTENGINE	Required for use of Agresso Report Creator and Excelerator.

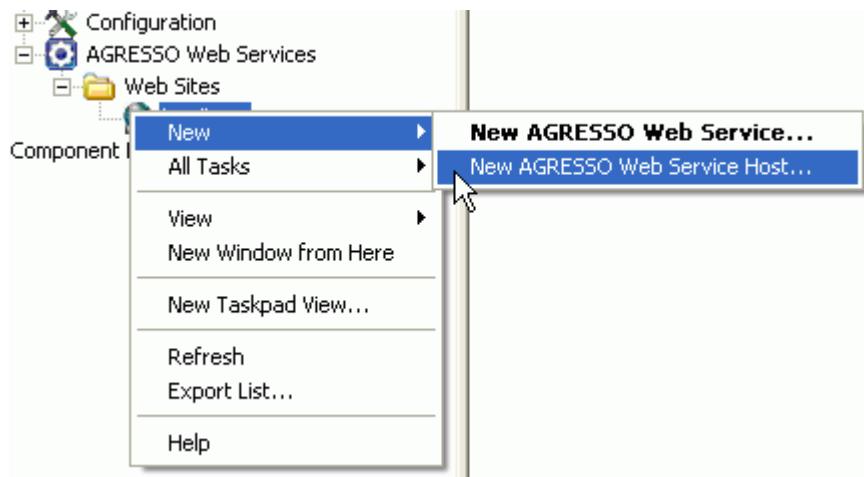
External services

External services come with a specific version postfix (e.g. Timesheetv200804), indicating that they may exist in more than one version. The available external services will be listed when you try to add them in the wizard.

Adding a Web service - general procedure

To add a Web service you must:

1. Open AMC and log on to the correct data source.



The internal services are added one at a time. For this purpose, you use the first option - **New Agresso Web Service**.

Note: You use the same option to add the Self Service client!

For external services, you will use the **New Agresso Web Service Host** option. The immediate difference is that it lets you add and publish several external services in one go.

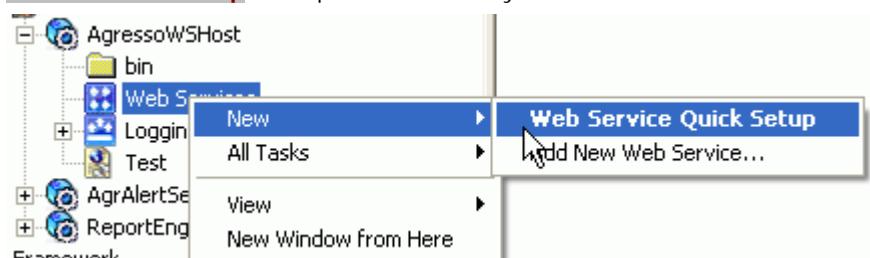
2. Right click on a **Web Site** node and make your selection.

A Wizard will guide you through the remaining steps.

3. Follow the instructions on screen.

When finished, the Web service - or Web Service Host - will be listed in AMC.

3B External only: If you selected Web Service Host, you must now add the external services you want: Right click on the Web Services node under the **AgressoWSHost** and select **New | Web Service Quick Setup** - and pick the services you want.



4. Click on a web service to open the service's management page.

Enable windows authentication

By default, users will get access to Agresso Web services through user name and password. If you want to enable Windows authentication, you must do as follows:

1. Select **Windows Authentication** as the preferred authenticator for the Agresso installation. You use the **Authentication Setup** window in ABW Smart client for this purpose.
 2. Click the **Configure windows authentication** link (the **Authentication** tab) in the properties window for the Web service (see Self service example below).
- Note:** This requires that the Agresso users are linked to a domain user. This is configured in **User Master File** (ABW Smart client).

The screenshot shows the Agresso - AGRESSO Self Service interface. At the top, there is a blue header bar with the title "Agresso - AGRESSO Self Service". Below the header, there is a navigation menu with tabs: "Application", "Configuration", "Test", "Overview", "Publish", "Document Archive", and "Authentication". The "Authentication" tab is currently selected. Under the "Actions" section, there is a button labeled "Configure windows authentication" with a small icon of a person. Below the button, there is a description: "Configure Integrated Windows authentication support for this application".

See [Authentication](#) for more details on the authentication setup.

Setting up a Web service for a second data source on the same server

Web services in installation directory

When the Web services were installed, they were each given a separate folder under *Web Services* in the selected installation directory. When you add (and publish) a Web service, AMC will generate a *web.config* file for the selected service, and save it in the service folder.

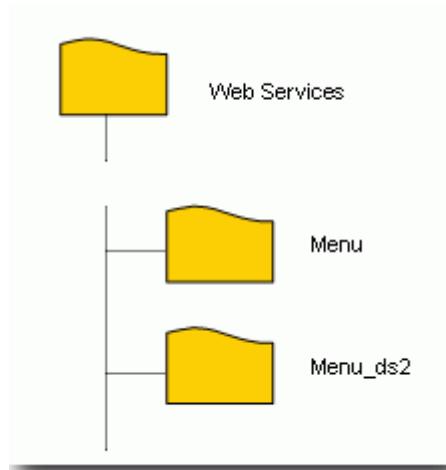
Adding Web service for a second data source

If you need to add a Web service to a second data source (a service that is already published for one database), you need to create a new service folder, with the correct files, before you can use AMC. Otherwise, AMC will overwrite the existing config file.

You must do as follows:

1. Copy the service folder you need and give it a meaningful name.

Example: We have made a copy of the *Menu* service folder, and called the new folder *Menu_DS2*:



2. Start AMC, log into the (second) data source and select **Add a New Agresso Service** (see **Adding a Web service - general procedure** above).
3. Move to the **Select Root Folder** dialog and browse to the folder you just created (e.g. *Menu_DS2*).
4. Complete the wizard.

Result: You have now successfully added a web service for a second data source!

General System Configuration

SYSTEM COMPONENTS OVERVIEW

Components overview

When we refer to System components in this context, we mean components that can be installed or configured from the Agresso Management Console - or other Agresso tools, and which is not covered by any previous topic.

These components are:

- Mail system
- SMS
- Printer plugins
- Help Configuration
- Document Archive
- Miscellaneous
 - Localization
 - License

Mail Configuration

MAIL CONFIGURATION OVERVIEW

AMC

You will work with the **Mail Configuration** node the **Agresso Management Console**, to handle mail configuration for the Agresso Business Server and users of the Agresso Smart Client.

Configuration elements

During mail configuration, you will work with the following basic elements:

- *Mail system integration plug-ins*. A mail system configuration is dependent on at least one mail system interface plug-in, ensuring correct communication between your existing mail system and Agresso. Currently, Agresso comes with four mail system interface plug-ins (see below). You can also add your own.

To set up a plug-in for use, you need to define one or more *Configuration profiles*, which implements the mail protocol with a default configuration.

- *Configuration profiles*. A mail system configuration profile is a set of rules and parameter settings for an integration plug-in. It holds two *profiles*, one for client users (*User profiles*) and one for mail from server side programs (*System profiles*).
- *User profiles* and *System profiles*. If the configuration profile shall be used for server side mail, you must always define some details for the System profile - or confirm the use of the default settings. User (client side) profiles will automatically be configured first time users try to send mail, based on the configuration profile they belong to.

Agresso Mail system interface plug-ins

Agresso is currently delivered with four mail system plug-ins. Three of these integrate with a mail protocol, while the fourth is a mail queuing solution (Agresso Mail Queue), that monitors the mail sending process executed by one of the others. It supports re-send options, error handling, status inspection and so forth.

AgSMTP

AgSMTP (SMTP) is the recommended choice for all server side mail. It should be set up as the mail executor (mail protocol), monitored by Agresso Server Queue.

AgMAPI

AgMAPI (for Simple MAPI compatible mail clients). For users with MAPI compatible mail clients on their desktop. Can not be used for server side mail.

AgrExMAPI

AgrExMAPI (Extended MAPI). AgrExMAPI handles server side mail, but can currently not be used as the executing system for Agresso Mail Queue.

AgrMailQueue

AgrMailQueue (Agresso Mail Queue) will not intermediately send the mail but stores the mail in a register handled by Agresso Message Server (AMS). The AMS server queue uses one (or more) of the other plug-ins to actually send mail, and the mail will be kept in the database until it is successfully sent. AMS supports re-send options, error handling, status inspection and so forth

Custom mail system integration plug-ins

Agresso Mail configuration also supports custom mail system plug-ins, provided that the plug-in is compatible with some important interface requirements. These requirements are currently not published, but can be obtained from Agresso consultants.

Example: configuration elements

A typical example is shown below of a mail configuration as it is displayed in AMC. There are two integration plug-ins in use, AgrSMTP and AgrMailQueue. We also find two configuration profiles based on the AgrSMTP template, while AgrMailQueue is the basis for one configuration profile:

Plug-ins and configuration profiles

Each plug-in provides you with a defined set of configuration options which will be displayed in the right hand window when you click on the mail system implementation node. A configuration profile allows you to set default rules and parameter values for users that later will be attached this profile. Below we show an extract of the configuration options provided by AgrSMTP:

If you select an AgrMailQueue configuration profile node, you find another set of options:

Profiles

User Profiles

When you click on the **User Profiles** node in AMC, all users registered with the active configuration profile will be listed in the right hand panel. This will be empty immediately after creation of the configuration profile.

When the users defined with this profile first try to send mail, they will be prompted to enter (or confirm) a few values - if that is allowed in the configuration profile.

System Profiles

If your implementation shall support server side mail, you will need to create at least one system profile that will be the identifiable mail sender. The system profile must match a registered user on the mail server.

Supported mail systems

Agresso currently supports the following mail systems:

Mail System	Description
SMTP (Simple Mail Transfer Protocol)	<p>For server-side mail integration, like distributing reports on mail from the Agresso Business Server.</p> <p>Smart Client users: SMTP does not have a user interface. When Smart Client users are set up with an SMTP profile, Agresso has provided a mail interface contained in the library <i>AgrMailUI.DLL</i></p> <p><i>AgrMailUI</i> contains an Agresso integrated Address Book and a</p>

	Compose Mail dialog with text and HTML capabilities
Simple MAPI (Messaging Application Programming Interface)	<p>Outlook Express and Windows mail implements Simple MAPI only.</p> <p>Simple MAPI is also supported by other messaging applications like Lotus Notes and Novell Groupwise.</p> <p>Note: Simple MAPI is <i>not recommended</i> for server-side mail integration.</p>
Extended MAPI	<p>Provides an extensive set of functions used to create mail-enabled applications. The full function library is known as MAPI 1.0 or Extended MAPI. Extended MAPI allows complete control over the messaging system on the client computer, creation and management of messages, management of the client mailbox, service providers, and so forth.</p> <p>Microsoft Outlook supports Extended MAPI.</p>

AgrMailUI - the Agresso provided user interface for SMTP users

Agresso Mail User Interface provides a user interface for Smart Client users that are set up with SMTP as mail profile.

Dialogs

AgrMailUI provides the user with two dialogs:

- A standard **Send mail** dialog - with functionality found in most mail clients.
- An **Address book** - integrated with the Agresso **User master file**.

Agresso table for contacts - *aagcontacts*

Using AgrMailUI, both personal contacts and global contacts are added to the table *aagcontacts*.

Global contacts: Global contacts are added with an asterisk (*) in the user_id column, meaning that the contact will be available for all Agresso users.

Personal contacts: When a personal contact is added, the User Id of the user making the registration are added in the user_id column, meaning that the contact only will be available to this user.

Mail addresses

The different mail systems handle mail addresses in different ways, depending on the mail client/mail system they integrate to. For instance, the SMTP plug-in will always expect SMTP addresses when sending mail. All other address types will lead to an error during the send operation.

The address book returns the mail address in the following formats based on the mail system used and the mail application:

Mail system	Mail client		
	Outlook	Outlook Express	AgrMailUI
Extended MAPI	SMTP	-	-
Simple MAPI	MS Format	SMTP	-
SMTP	-		The format that is used when the address is stored (normally SMTP format)

Valid address formats

As a general rule, all mail addresses should be stored in SMTP format. The following address formats are all valid:

[Display Name]SMTP:mail.address@mycompany.com

[Display Name]mail.address@mycompany.com

SMTP:mail.address@mycompany.com

`mail.address@mycompany.com`

The Test node

You can use the **Test** node to check the setup for the various mail systems.

Known problems

- SMTP: When you select NTLM authentication, you must ensure that the Business server (configured in AMC) logs on as a network user known to the SMTP server. (See [Initialize and Maintain the Business Server Environment](#).)
- Sometimes a Server Busy message is appearing after the **Send** button is clicked when using the AgrExMAPI mail system. This might be annoying but does not lead to any errors.

Server Busy Message

- If Outlook is your default mail client and Simple MAPI is used as the mail system, then the addresses returned from the address book will be in a wrong format (MS Exchange format). You will only be able to send mail from Agresso with E-mail addresses in standard SMTP format. If you are using Outlook as mail client, then please make sure that Extended MAPI is used as mail system.
- When the mail dialogs are started from the Agresso Smart Client (in some versions of Outlook), problems may occur. For instance, when using Outlook 2000, you will not be able to open the address book. When using Outlook XP (2002), you will need Service Pack 3 (or later) installed. If you have an earlier version of Outlook XP installed, you may find that the Smart Client crashes when a mail dialog starts (i.e. tries to start!..)
- When opening a mail dialog, you may find the following error message:

Error during SMTP login (Connect): Could not connect to the SMTP server '10.42.8.12' on port 25, Error(10053): Failed to connect client socket. An established connection was aborted by the software in your host machine.

The reason is most likely that port 25 is blocked on the local computer, since network administrators often do prevent "mass mailing worms" from sending mail.

CONFIGURATION TASKS

Mail Configuration

The main configuration tasks are as follows:

1) Select mail system integration plug-ins

You will first add the plug-ins you need, based on your existing mail infrastructure.

Note: Agresso Mail Queue requires that at least one SMTP system also is set up, since it will need an SMTP system to actually send the mail. Agresso Mail Queue can also direct mails to more than one SMTP system - if any errors occur in the first one.

Selected mail system integration plug-ins will appear as new nodes in AMC.

Next, you will add configuration profiles for the new nodes, and set up user profiles and system profiles. AMC will add a new node for each configuration profile, each with the sub nodes User profiles and System profiles. An example is shown below:

2) Define system profile(s)

To make a valid system profile, you must right click the system profile node and create a new profile.

2 B.) Server side mail only: Activate the mail system

When the Configuration profile and system profile is in place, you link it to the Business server by using the *System profile name*. You will use the **Business Server** node in AMC and select the mail profile under **System Integration**. The Business server will then be ready to send mail.

3) For Agresso Server Queue: Define Queue settings

To complete the setup of an Agresso Mail Queue, you must link the registered SMTP system profile to the queue. You use the **AMS** node, found under **Business Server | Server Queues**.

Client users configure themselves

All Smart Client users trying to send mail, will automatically be presented with a mail setup wizard, prompting them to configure the available user profile, and thus creating specific User Profiles for themselves. The options will be defined in the Configuration profile.

A system administrator will find these profiles under the **User Profiles** node, where they can both be edited and deleted.

CREATING A CONFIGURATION PROFILE

Introduction

General

Depending on whether your mail system is intended for system reporting, for ordinary use by employees, or both, you must implement at least one configuration profile. Regardless of the number of profiles you need, and the eventual user types intended for the profiles, you follow the same basic steps for each.

Your starting point is always the **Mail Configuration** node in AMC.

Business server restrictions

A mail system (intended for the Business server) should as a rule be based on Agresso Mail Queue, but SMTP or Extended MAPI can also be used.

Extended MAPI: If you will use an Extended MAPI template, **Microsoft Outlook** must be installed on the server and configured for the account to be used by the Business server.

Main tasks

Creating a new Configuration profile, involves four small tasks:

- A. Select a mail system integration plug-in and add it as a new node to AMC.
- B. Select the new node and add a new Configuration profile node to AMC.
- C. Select the Configuration profile node and define the default setup on the Mail Configuration page
- D. Add at least one System profile.

Create a plug-in node

A. Add plug-in

1. Right-click the **Mail Configuration** node and select **New | Add Mail Integration Plug-In**
2. Select the desired plug-in and click **Finish**

As a result, the AMC will be expanded with a new node:

B. Add configuration profile

The next step is to add the configuration profile (the system and user profiles come automatically):

1. Right-click the new plug-in node and select **New | Create Configuration Profile**.

A dialog box is displayed where you can enter a descriptive name for the configuration profile.

2. Enter a meaningful name and click **Finish**.

A new configuration profile node appears, with three sub-nodes, named **User Profiles** and **System Profiles** and **Test** respectively.

C. Configure the configuration profile

1. Select the node to view the different configuration options. You can control which settings the user should see, and which settings the user should be able to change.

[Page details](#)

Descriptions

You set default parameter values for **Authentication**, **Mail Options** and **SMTP Server Connection** (in this example). The various check boxes are used to set rules for client users, when they try to send Agresso mail for the first time.

Authentication (SMTP)

AgrSMTP supports the following authentication methods:

- **No login** - SMTP server looks at your IP address to ensure you are on the RCN network.
- **Authentication Login** - The authentication data is sent to the SMTP server as Base64 encoded text.
- **NTLM (NT LAN Manager)** – Uses Microsoft Authentication Protocol to authenticate the user (Agresso user or server side software) trying to access the SMTP server to send mail.
Note: To be able to use this protocol when sending mail from the Business Server, assure the Business Server service logs on as a valid network user.
- **Cram MD5 Method** - CRAM-MD5 is a challenge-response authentication mechanism to authenticate the SMTP connection. It encrypts the client's user name and password and is more secure than Plain Login and Authentication Login.
- **Plain Login** – The authentication data is sent to the SMTP server as plain text.

Default Mail Encoding

The value selected for **Default Mail Encoding** must correspond to the value given by Agresso system parameter DEF_FILE_ENCODING. The following two values are most relevant:

If DEF_FILE_ENCODING = **ANSI**, **Default Mail Encoding** shall be set to **Western European (ISO)**.

If DEF_FILE_ENCODING = **UTF8**, **Default Mail Encoding** shall be set to **Unicode (UTF-8)**.

2. Change the values to fit your local environment.

Example: When setting up an SMTP mail integration you typically set which type of Authentication the mail server is using as well as the SMTP server's host name or IP address where the mail server can be reached. We recommend that you fill in the fixed values for your environment and select **Read Only** and **Hide for User** for these values.

There is no save option as all values are stored continuously.

D. Add a system profile

If the mail configuration profile is to be used from the server (the **Not Empty** box is checked - see Page example above), you need to create a system profile.

1. Right click the **System Profiles** node and select **New | Add Mail Profile**

2. Type in the system user name and click **Next**.

Note: This is the name you will use when you later link the implementation to the Business server.

3. Type in the Display name (for the email address) and the email address to use, then click **Next**.

4. Set the correct parameters and you are done.

Note: It is mandatory to have a correct mail account name and password.

Result

You have now created a new, general configuration profile, in principle available both for Smart Client users and for the Agresso Business Server.

ACTIVATING A CONFIGURATION PROFILE

Remember

Only SMTP and Extended MAPI are supported as mail systems for Agresso Business Server! See [Creating a Configuration Profile](#).

Agresso Server Queue

If you want Agresso Server Queue to handle your mail, you must link the System profile for the *server queue configuration profile* to the Business server.

Two tasks

To activate a configuration profile, you must

- Create (and configure) at least one System profile for the configuration profile. Later, the configuration profile will be identified by the System profile (system user) name.
- Link the configuration profile to the Business server.

Activate a new configuration profile

Create the system profile

When your configuration profile is in place, you can proceed with creating the actual system profile for the solution.

1. Right-click the **System Profiles** node and select **New | Add Mail Profile**.

You are presented with a simple dialog box.

System Profile Name

2. Enter a meaningful name and click **Next**.

If the profile is based on SMTP, you will have to identify the system mail user. Otherwise, continue with step 4.

3. If SMTP based: Enter the display name and the Business server's mail address (the From address for mail sent from the Business Server processes), and click **Next**.

4. Enter or modify the configuration and click **Next** when completed.

The new system profile appears in the right panel in AMC, ready to be used.

Link the profile to the Business server

To link a system profile to the Business server, you start with the **Business Server** node in AMC.

1. Select the **Business Server** node to open the configuration window.
2. Select the **System Integration** tab and then the [Mail profile](#) link.

Example

3. Select the (newly created) system profile from the **Mail Profile** drop-down list.

The system profile is ready to be used by the Agresso server reports.

Additional option for SMTP profiles

If you select **Enable** in the **Send on behalf of** drop-down list, users who order server reports will appear as the mail originators.

Logging options

When a server process sends mail, the main actions will be written to the default log file.

Log Level

To get a more detailed log, you can change the **Log Level** to **Add queries**. All mail steps and error messages will now be logged.

SETTING UP AN AGRESSO MAIL QUEUE

Main tasks

To set up a new Agresso Mail Queue, you must do as follows:

- Create an SMTP configuration profile, as described in [Creating a Configuration Profile](#).
Note: You shall not activate this profile when using Agresso Mail Queue. The *Agresso Mail Queue* is the link to the Business server.
- Create an Agresso Server Queue according to same lines as above. Continue with [Activating a Configuration Profile](#).
- Identify the SMTP implementation(s) that shall be managed by the mail queue.

Create an Agresso Mail Queue

Select mail system integration plug-in

1. Right-click the **Mail Configuration** node and select **New | Add Mail Integration Plug-in**
2. Select the desired plug-in and click **Finish**

Add configuration profile

The next step is to add a new configuration profile:

1. Right-click the new plug-in node and select **New | Create Configuration Profile**.
2. Enter a meaningful name and click **Finish**.

A new configuration profile node appears, with two profile sub-nodes and a **Test** node.

Configure the configuration profile

Select the node to view the different configuration options. You can control which settings the user should see, and which settings the user should be able to change.

Identify the configuration profiles that shall be managed by the mail queue

AMS node

You will use the AMS node to handle the relations between the queue and the configuration profiles that shall be managed by the queue.

Parameter settings

To set up a mail queue, you will run the Add Mail queue Profile wizard. It will lead you through two or three screens:

- A. When you are prompted to select mail profile, you must select a system profile for the SMTP configuration profile that will be used by the queue.
- B. Next, you will enable or disable the *Reset interval*.
- C. If you have enabled a Reset interval, you have to enter the maximum number of temporary errors to be accepted, before the queue will stop trying to re-send it. In addition, you will set the time between each retry.

Procedure

1. Click on the **Mail Queue View** node under AMS and click the [Initialize](#) link under **Configuration / Profiles**:

A Wizard will help you set up the Server Queue parameters.

2. Run through the wizard to complete the initial Agresso Server Queue setup.

Result: As a result, you will be presented with new version of the Profiles page, allowing you to add more configuration profiles to the queue (the [Add mail profile](#) link).

SMS Configuration

CONFIGURING SMS

SMS Configuration

General

You configure the SMS settings in much the same way as mail. You will need to select the appropriate SMS system (plug-in), and then create the Agresso profile(s). The **SMS Configuration** node in AMC gives you access to the necessary configuration tools.

The configuration is stored in the table *aagsystemconfig*.

SMS and AMS

In version 5.5 the only part of the system that integrates directly to the SMS plug-ins is the Agresso Message Service (AMS) server queue running as part of the Business Server. All SMS needed to be sent from the different parts of Agresso will be posted to the AMS through the database. AMS picks up new messages and will try to send or resend previous failed messages using specific rules.

A prerequisite is that an arrangement with an SMS supplier supported by Agresso is reached and all software required by the supplier is installed and running.

Supported SMS systems

Agresso 5.5 supports two SMS systems:

M:Science SMS Server

Enables organisations to provide two-way SMS messaging facilities to their personnel and corporate information systems, using existing messaging infrastructure.
Agresso delivers a plug-in named **AgrSMSMSScience** for this.

Note: *AgrSMSMSScience.DLL* is found in the Agresso *bin* folder, and you will need to browse for this file if you select M:Science SMS Server as your SMS system.

Agresso SMS Demo

An SMS solution well suited for demonstration purposes and small scale operations.
Agresso SMS Demo requires a cell phone attached to a COM port on the computer running Agresso Business Server.

More details are found below.

M:Science SMS Server (*AgrSMSMSScience.DLL*)

Description

See <http://www.m-science.com> for a detailed product description.

The SMTP Gateway is a part of the M:Science implementation and acts as a bridge between any SMTP compatible messaging system and the Server component. The Agresso M:Science SMS plug-in builds on top of the SMTP Mail plug-in (AgrSMTP.DLL). It takes care of all message formatting and uses SMTP to send the messages. Based on how M:Science inbound messaging is configured, the SMTP messages reaches the M:Science SMTP Gateway, which does further processing to get the SMS sent.

Parameters

The following parameters are required:

Parameter	Description
Telephone field	When the system parameter IN_MOBILE_FIELD is off, Telephone field number is used to get the mobile telephone number from the address

number	record sent to the plug-in (from AMS). If system parameter IN_MOBILE_FIELD is on, AINAPS will use the value of this parameter (default is 4).
Mail Drive Name	The Mail plug-in to use when sending mail to the SMTP gateway. Locked to AgrSMTP .
SMTP Gateway	
Virtual Domain	Use of the SMTP Gateway requires the definition of a <i>virtual domain</i> for SMS messaging traffic. Example: <code>send.sms</code> or <code><company name>.sms</code> This 'virtual domain' is only relevant within the organization concerned and has no relation to your own internet domains. You should choose a domain that does not exist on the internet because it is used to identify SMS traffic within your organization. SMS messages are then addressed as <mobile phone number>@<virtual domain>, for example <code>0777700001@send.sms</code>
From Name for mail to SMTP Gateway	The name of the mail sender.
From Address for mail to SMTP Gateway	The SMTP mail address of the mail sender
Mail Server Authentication Method	How to validate the SMTP connection
SMTP Server Connection (required by AgrSMTP)	
Host Name/Address	Computer name or IP address of the SMTP server
Port Number	Port Number of the SMTP server
Automatically connect to internet	Automatically do dial-up if internet connection is not activated.
Local Bound Address	Can be "Any IP Address" or <code>127.0.0.1</code> which is the local IP address
Mail Interface	Not used for SMS

Agresso SMS Demo ([AgrSMSDemo.DLL](#))

Agresso SMS Demo is an SMS solution well suited for demonstration purposes and small scale operations. The Agresso SMS Solution requires a cell phone attached to a COM port on the computer running Agresso Business Server.

The following parameters are required:

Parameter	Description
Telephone field number	When the system parameter IN_MOBILE_FIELD is off, Telephone field number is used to get the mobile telephone number from the address record sent to the plug-in (from AMS). If system parameter IN_MOBILE_FIELD is on, AINAPS will use the value of this parameter (default is 4).
COM Port Settings	
COM Port	This is the COM Port the phone is attached to. Use the format COMn, where COM is the port prefix and n is the COM port number. Example: <code>COM1</code>
Default Timeout in milliseconds	Send timeout when the plug-in is sending commands to the mobile phone. If you have problems sending SMS messages, try to increase this value to 10000 milliseconds or higher.

Validity Periods	
Use Validity Period	If turned off, the default value set by the service center is used.
Default Relative Validity Period	Default time the message center will try to deliver a message when the mobile phone is not available.
Connection	
Service Center Phone Number	Phone number to the SMS Service Center
Service Center Phone Format	This is the Type-of-Address format of a phone number. A phone number in international format looks like 46708251358 (where "46" is the country code). In the national (or unknown) format the same phone number would look like 0708251358 . The international format is the most generic, and it will be accepted also when the message is destined to a recipient in the same country.
Default Destination Phone Format	Note: For the demo solution, select a destination phone format and stick to it when entering destination phone numbers in Agresso (see Service Center Phone Format above for details.)
Require Status Report	Turn on/off use of status report for send SMS. This value is returned by the service center. The return value is not returned by the demo solution.
Replacement Char	Default replacement character

Configure SMS

You must perform the following tasks to set up an SMS configuration from the Agresso Management Console:

- a. Select an SMS plug-in
- b. Create an SMS System User Profile with required configuration settings

Select SMS system

To select the SMS system, do as follows:

1. Right-click the **SMS Configuration** node, and select **New | New SMS Integration Plug-In**
2. Select the system and click **Next**.

Note: The **.dlls** for the plug-ins are located in the **bin** folder.

When the SMS system is selected, you can create the new SMS profile.

Create the SMS Profile

To create the SMS Profile, proceed with the following steps:

1. Right click the (newly added) SMS system node in AMC, and select **New | New SMS Profile**.
2. Enter a meaningful name and click **Finish** to create the profile.

The configuration information is now displayed when you select then (new) profile node window. To complete the configuration task, you must fill in the required parameters. See Available SMS Systems for details.

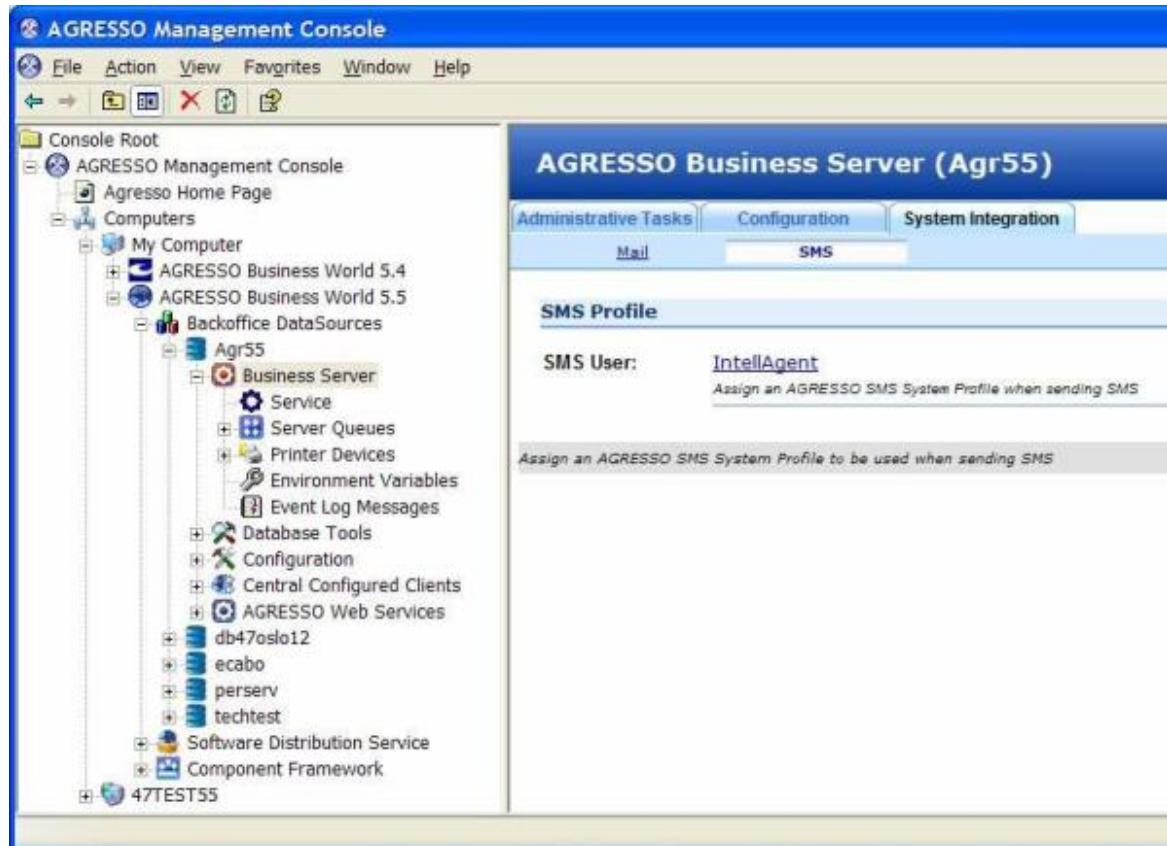
3. Fill in the required parameter values.

Changes will be automatically saved. The SMS profile is now ready to use.

Assign the SMS Profile to the Business Server

As with Mail Profiles, you must link the SMS profile to the Business Server or to a specific server queue. Using the profile information the server processes loads the SMS plug-in and set the session parameters to connect and send SMS.

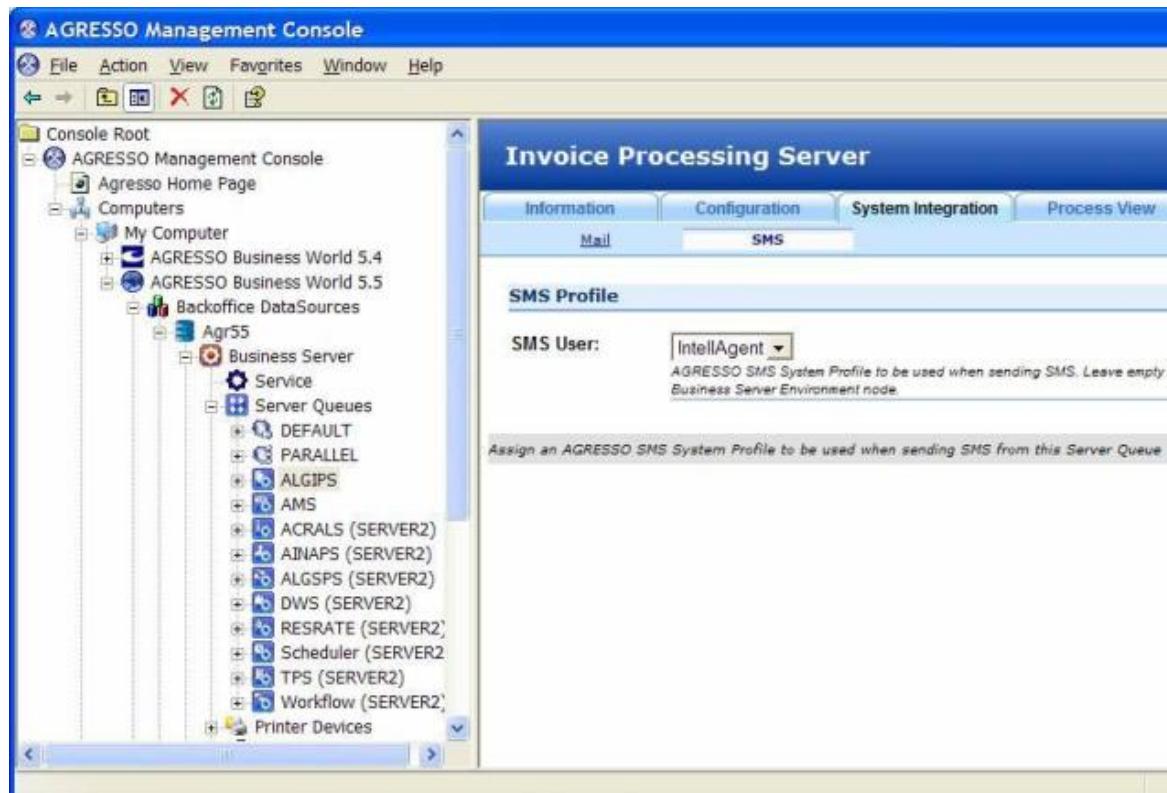
Assigning Profile to Server



As with mail, you can choose to enable SMS only for a given server queue, or use different SMS profiles for different server queues.

! In version 5.5, the only process capable of sending SMS is the Agresso Messaging Service (AMS) running as a server queue.

SMS from Server Queue



Test the SMS connection

For the moment, there is no test button available. The way to test the SMS connection would be to set up IntellAgent for SMS and then generate an event.

SETTING UP MAIL AND SMS ALERTS

Overview

Error alerts makes it possible for the Agresso Business Server service to send an alert when a report or process fails. The only prerequisite is that there are relevant (Mail and SMS) system profiles available.

You can configure different alert settings for different error types. If you configure settings on the server queue level, these settings will override the settings on the business server level.

You find the configuration options on the **System Integration** tab on the **Business server** node (or server queue node).

Error types

The Business Server service receives a return value from executed processes. If the return value indicates that an error has occurred, the service will check if alerts are enabled for the error code, and if so, it will send the alert to the target address (mail address or phone number).

You can configure up to four different alerts, one for each of the following error types:

Error type	Description
TECHNICAL ERROR	Indicates general error that the administrator should look in to. Examples: The database connection is broken or the process fails to run.
FUNCTIONAL ERROR	Indicate that required data is missing, making it impossible to complete a report or a process..
NO ROWS	Occurs when a report returns zero rows. Note: This is not really an error; there is just not any data. If you set up an alert for this error, it's recommended to only send it to the report owner.
OTHER ERRORS	Covers all other errors, for instance if a process is killed in AMC.

Configure Alerts

Options overview

The configuration parameters (fields) are shown in this example of the **System Integration** tab:

Error Type

Error type: TECHNICAL_ERROR
Configure settings for technical errors. This is a serious error. It's recommended sending alerts both to administrator and the user that ordered the report.

Mail Alerts (TECHNICAL ERROR)

Enable E-mail alerts: Send mail alerts

E-mail address: asmund.kristiansen@agresso.no;testuser@agresso.no
Alert will be sent to this address. Multiple E-mail addresses can be added separated by semicolons

Send E-mail to report owner: Send an E-Mail alert to the user that ordered the report

Test: Test mail

SMS Alerts (TECHNICAL ERROR)

Enable SMS alerts: Send SMS alerts (It's recommended that the AMS server queue is enabled)

Phone number: +4748012771
SMS will be sent to this number

Available time: Always
Select a period for when you want SMS alerts to be sent

Test: Test SMS (AGRESSO Message service is not used when testing SMS)

Configure alerts to be sent to administrator when reports fails. You can configure different settings for the different error types

Configure Alerts

Do as follows:

1. Open the Administration window for the item you want to add alert(s) to (meaning: click on the **Business Server** node or a **Server Queue** node).
2. Select the **System Integration** tab and verify that the system to configure is attached to a profile. You do this by checking the settings when you click the [Mail](#) and [SMS](#) links respectively.
3. Click the [Alerts](#) link to open the configuration page for alerts.
4. Set up the Alerts, and click the [Send ...](#) button(s) to check the result.

Alerts in the Registry

Registry locations

All the mail configuration settings are stored in the registry of the server you are configuring, at the following location:

Business Server level:

`HKLM\SOFTWARE\Agrezzo\Platform 2.2\Data Sources\<datasource>\Business Server Environment\Admin Alert\<error type>\<media>`

Server queue level:

`HKLM\SOFTWARE\Agrezzo\Platform 2.2\Data Sources\<datasource>\Business Server Environment\Server Queues\<Server queue>\Admin Alert\<error type>\<media>`

When locating the settings to use, the **Agrezzo Business Server** service will first look for the setting on server queue level. If none exists, it will look for settings at the Business Server level.

Troubleshooting

The two Test buttons allows you to test both Mail/SMS configuration and alert configuration. The AgrBusinessServer service must be running when testing mail and SMS alerts.

SMS test fails

If the SMS message appears to be sent (you get no error message), it is probably an SMS configuration problem that Agresso is unable to catch. Make sure the **Service number** in the SMS configuration is correct, and make sure that the language code is included in the phone number you will send SMS alerts to.

If the AMS server queue exist, it will be used to send the SMS. To make AgrBusinessServer always send the SMS alert directly, the registry value "Use Message Queue" (DWORD) can be created and set to 0. (see "Alert settings in registry" section)

Print Configuration

PRINT CONFIGURATION OVERVIEW

Introduction

This topic is limited to a few details about plug-in management. We describe how you can add a plug-in for Crystal Reports, and how you set a default plug-in for a report type. This latter subject is currently not relevant, as no alternative plug-ins exist.

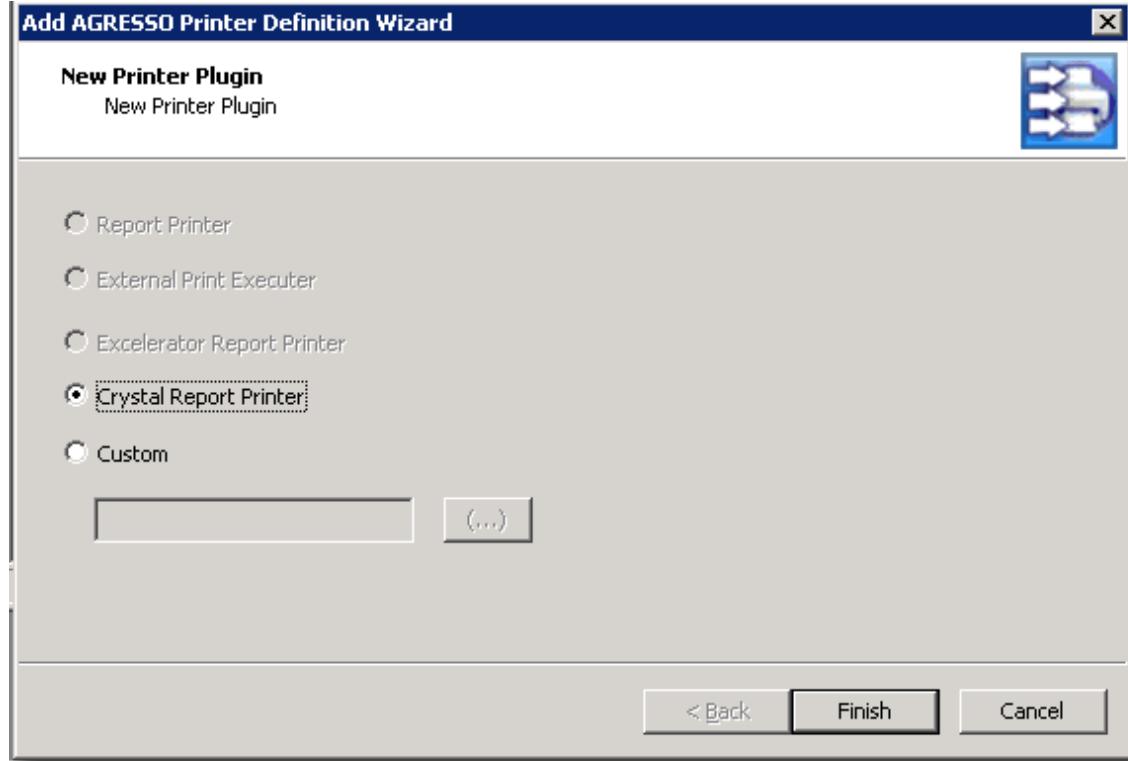
Please see [Printer Setup in Agresso](#) for an explanation of Agresso printers and plug-ins.

Adding the plug-in for Crystal Report

Plug-ins' added are listed below the **Print Plug-ins** node. To add the Agresso Crystal Report Print Plug-in, you do as follows:

1. Right click the **Print Plug-ins** node and select **New | Add Print Plug-in**.

Result: A Wizard displays available plug-in options:



2. Select **Crystal Report Printer** and click **Finish**.

Crystal report runtime files must exist to be able to add the AgrPrintCrystal plugin. When printing crystal reports, make sure that the "Background processing" checkbox on the printer definition is checked, this will start a separate process that executes the printing.

Adding a custom plug-in

When adding a Custom plug-in, you will - in addition to the procedure described for Crystal Report - browse for the .dll containing the plug-in.

Managing document type associations

Possible inconsistencies

A Print plug-in can in principle support several document types, and, if you have added a new plug-in, you will probably find that printing of a certain document type is supported by more than one Plug-in.

In such situations, there is no way for **Agresso Management Console** to determine the default plug-in for a given report type, and AMC will simply select the first one that fits.

In such cases, you must explicitly set the default association between a document type and the plug-in to use.

Set default plug-in for a Document type

To select a default Print plug-in for a given document type, you select the **Document Type Association** node and use the drop down list to set the default Plug-in.

Help Configuration

WEBHELP CONFIGURATION

HTML based help system

Agresso WebHelp is an html-based help solution compatible with all major browsers: Internet Explorer, Netscape, Firefox, Opera, etc.

Installation path

General

Agresso WebHelp must be installed on a Web server, available for all users. See [WebHelp Installation](#).

Language

There may be several language variants of the Help system, located in separate folders.

Examples: The URL for the English (UK) version can be: <http://<Web Server name>/SCHELP/EN/SCmaster.htm>.

The URL for the Swedish version can be: <http://<Web Server name>/SCHELP/SE/SCmaster.htm>.

Note: Make sure you remember the server name as well as the virtual path (URL) to the Agresso Web Help system!

Configuration overview

To configure the help system, you will use the **Help Configuration** node (under the **Configuration** node) in the Agresso Management Console.

Help configuration information is stored in the database, and you must therefore log on to the database before you can proceed.

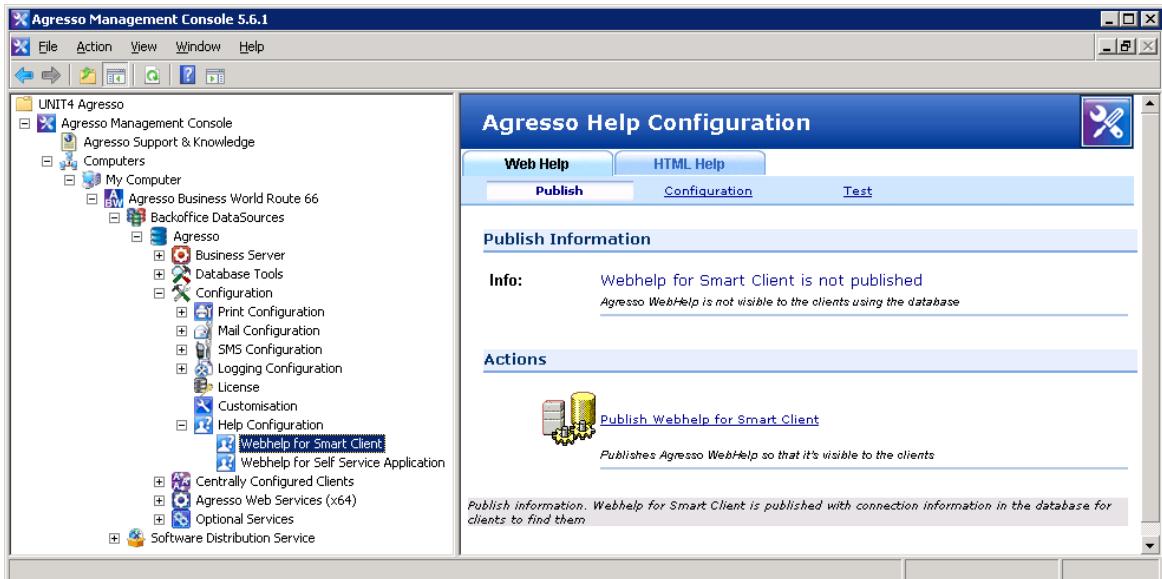
The point is to publish the Web Help so that clients will discover it whenever a user asks for Help.

Configure (publish) Agresso Web Help

Note: Log on to Agresso before you start!

1. Select the **Help Configuration** node to bring up the **Agresso Help Configuration** window.

 Example: Agresso Help Configuration



2. Select the **Publish** option on the **Web Help** tab and then [Publish Webhelp](#) link to start the **Publish Web Help Wizard**.

- 3.** Select the protocol to use (HTTP or HTTPS), and click **Next**.
- 4.** Select the server name where Online Help is installed and running, and click **Next**.
- 5.** Enter the base path or the name of the virtual folder given to the Online Help when installed, and click **Next**.

The Online Help is now properly published.

- 6.** Select the **Configuration** option on the **Web Help** tab.

The configuration is complete.

Test the configuration

You can select the **Test** node in AMC to check that the base path is correct (The language code of the Agresso user logged into AMC is used when creating the test URL path).

Change Web Help settings

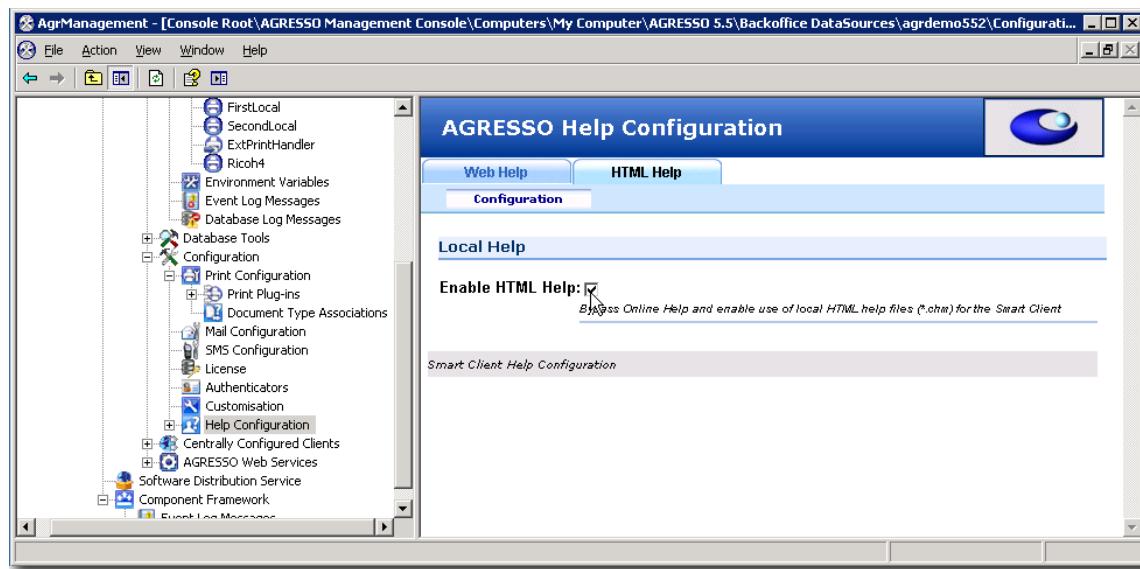
If you want to change any of the parameters set during publishing of Web Help, you must

1. Withdraw the service.
2. Configure (publish) Agresso Web Help - as above - once more.

HTMLHELP CONFIGURATION

Enable HTML Help

If you want to use the HTMLHelp option for On-line help (relevant for Smart Client users only), you must explicitly enable HTML Help in AMC:



System parameter: When you check Enable HTML Help, you actually set the value of the system parameter FORCE_LOCAL_HELP to 1. The Smart Client will then use the HTMLHelp version of the Help files.

Additional configuration when running ABW from a network share

If ABW has been made available through a network share, you must set a few (new) properties in Centrally Configured Client to ensure that the HTMLHelp works smoothly. See [Creating a Centrally Configured Client](#).

Note: The described configuration requires use of Agresso Software Distribution Server. If you need manual configuration, see http://www.helpandmanual.com/products_hhreg.html.

DOCUMENT ARCHIVE CONFIGURATION

Reference

The settings for the Document archive solution is managed from the Agresso Smart client.

Configuration issues

There are two configuration issues which may affect the general configuration:

- If one or more document types are configured to be stored in the database (AgressoBLOB storage), the table *adsfileblob* will be affected.
- If one or more document types are configured to be stored in the file system (AgressoFILE storage), it is necessary to set up a folder for temporary storage. You will use the Agresso Management Console for this task, see [Adding Agresso Self Service](#).
- You must use AMC to set a path for the environment variable AGRESSO_OCR. This will be used by the **DS04 OCR Export** server process. Details below.

Set path for AGRESSO_OCR

- Open the **Environment Variables** node (under **Business Server**) and check the default path for AGRESSO_OCR.

[Agresso_OCR example](#)



2. If desired, change the value, and save your settings.

Localisation Update Packages

Update Manager

When standard 5.6 installation is in order, you will need to use the **Update Manager** in **Agresso Management Console** to add the upgrades listed below.

- **Localisation:** You find the update file for your country (generalised as <country> in the file path below) in the following file:
ABW/Localisation/<country>/Update Manager Files/ <country>.update.dll.

Procedure details

See [Using the Update Manager](#) for details.

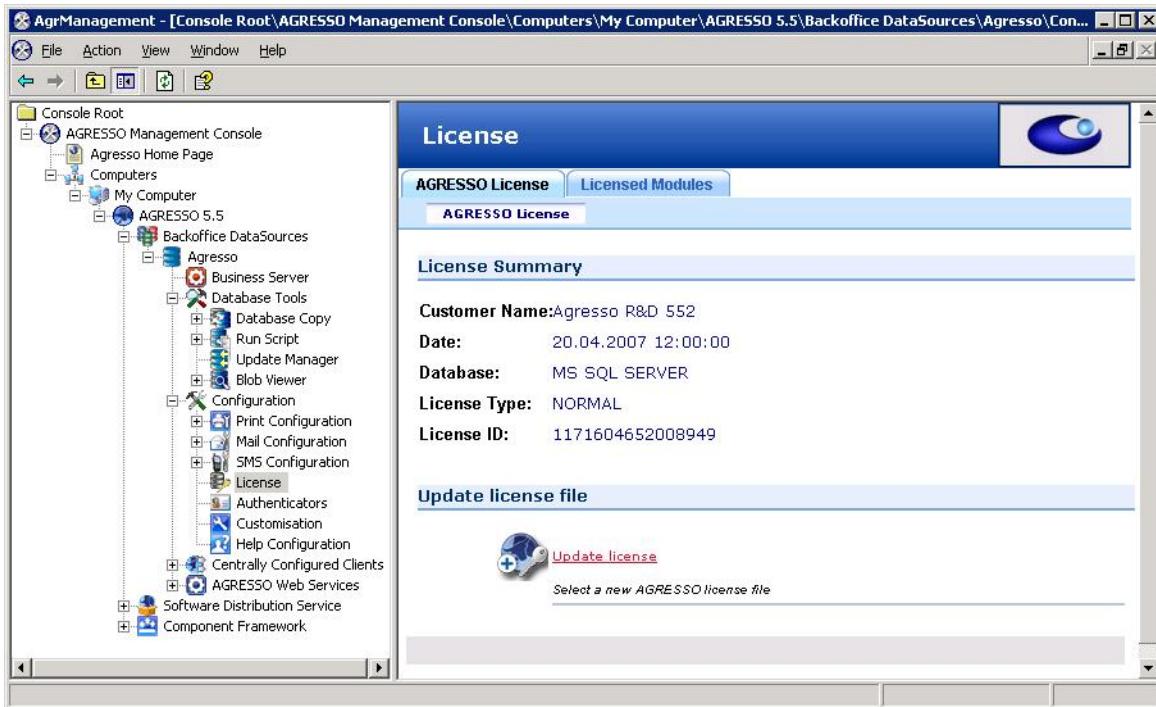
LICENSE

View License

The license node in AMC allows you to add a new Agresso license to the database. It also enables you to view the licensed modules and the number of users registered for each module.

If you have a demo license installed, this node will inform you if the license will soon expire, or if it has already expired.

 [View License](#)

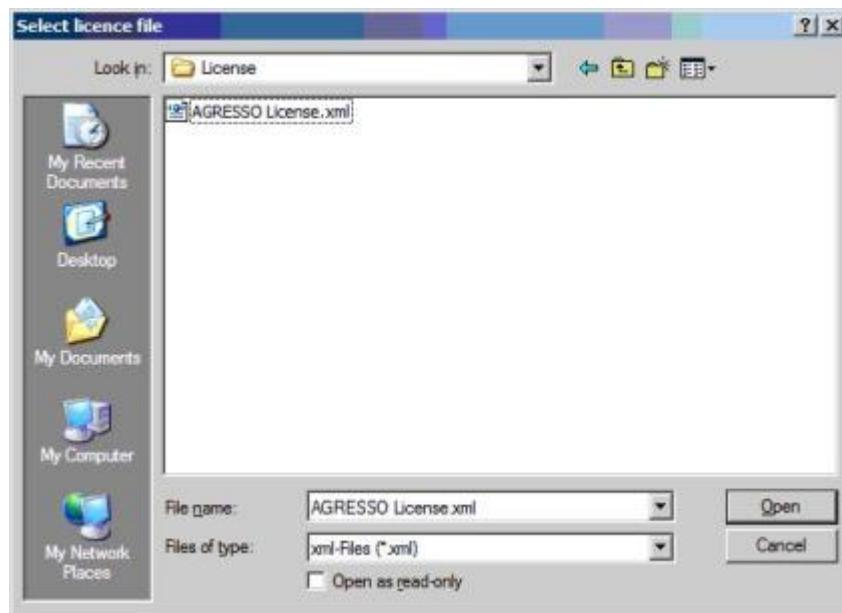


Add License

You will have received the new license by email. Before you can add the new license to AMC, you must have saved it somewhere on disk.

To add a license you can either right-click on the license node and select **Add License** from the **All tasks** context menu. Or you can click on the [Add License](#) or [Update License](#) link found in the right pane of the management console. Then you will be prompted to browse for and select a license file.

Add License



You can also add a license using the Agresso Client Configuration tool.

Upgrading Agresso

From Agresso 5.6.0 to Agresso 5.6.1

UPGRADING FROM 5.6.0 TO 5.6.1

Prerequisites

- The new version of **Agresso Business World** must have been successfully installed before you start the upgrade process.
- We assume that you have profound knowledge of the Agresso installation and the RDBMS used.

Prepare the upgrade

Backup the database

Before you run the upgrade wizard, you should back up your Agresso data. To be completely safe, you should also use the **Agresso Copy** program to make a complete copy of all database tables.

Verify disk space

Available disk space needed for the upgrade is approximately the size of the largest table + 10%.

See the list of table changes in the Appendix.

Create 5.6 Data Source and initialize Business Server environment

Use the **Agresso Management Console** (AMC) to create a new 5.6 Data Source connected to the old database.

When the connection is up and working, you must also initialise the Business Server environment (select the **Business server** node in **AMC**) and then **Initialise Business Server** .

Now you can run the upgrade wizard. See [Upgrading Procedure 5.6.0 to 5.6.1](#).

amendment tables

Logging

If amendment logging is turned on for an Agresso table, an amendment table (or shadow table) are created and continuously updated with all table changes.

Note: During upgrade, all amendment logging will be turned off (by the wizard), and switched on again when the convert script is complete.

Table for amendment tables

All amendment tables are defined in the table `aagamendlog`.

Naming standard: An Agresso table name is constructed from the structure `a<module><identification>`, while an amendment table is extended with the letters `shd` between `<module>` and `<identification>`.

If you turn on amendment logging for `acrclient`, the amendment table `acrshdclient` will be created and added to `aagamendlog`.

Recreate active logging triggers

If some of the shadow tables are not converted correctly, and the upgrade wizard is unable to fix the error in the database check. The shadow tables can be re-created by renaming or dropping the existing table, and then re-enable amendment logging for the table in the [Activation of logging server](#) (AG30) page in the smart client.

When errors occur

It is important that all errors encountered during the upgrade process are reported back to Agresso.

Customer

As a customer you are asked to report the errors as tickets in the Heat system.

Partner or subsidiary

If you, as a partner or subsidiary, have access to the Heat system, report the errors as Heat tickets. Otherwise, please report the errors to the Technical Forum - Agresso 5.6 Technical Consultants <http://forum.agresso.no/>.

Do not report the problem if it is related to duplicates in the database, full disk, full tablespace or similar.

Check user defined queries after upgrading

As part of the system upgrade, the ASQL parser has been improved with stronger syntax check and more consistent error handling. If a complex query contains an error, no part of the query will be executed. Instead, ASQL will report a parser error.

Previously, valid ASQL appearing before the error would have been executed, while the rest would be ignored. Now, the query will not run at all.

It is therefore important that you run all user defined queries after the upgrade.

UPGRADING PROCEDURE 5.6.0 TO 5.6.1

The upgrade wizard

You will use the wizard located in

...Agresso 5.6\DatabaseScript\DbUpgrade\UpgradeWizards.exe.

Start the Wizard

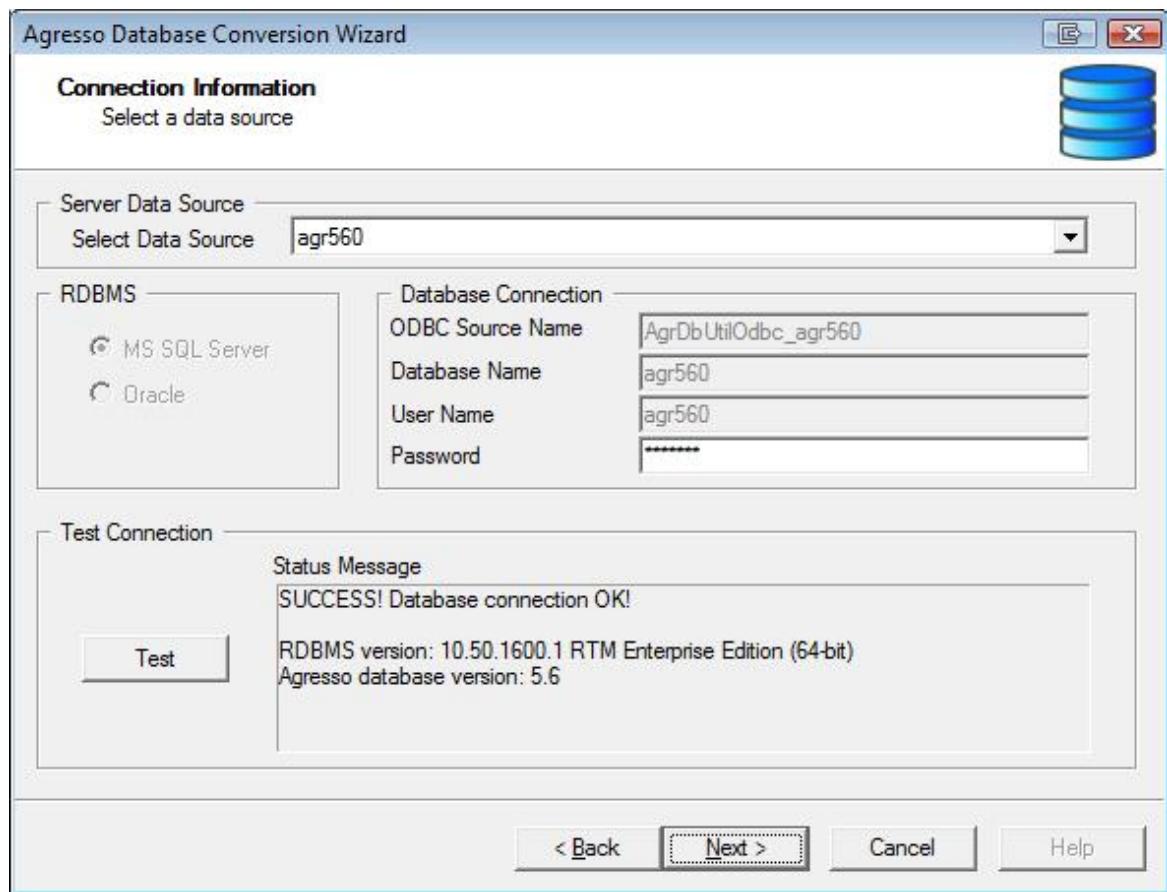
- A. Start *UpgradeWizards.exe*
- B. Select **Convert from 5.6.0**

Procedure

1. When the upgrade wizard is up and running, select Step 1, *Main Upgrade Wizard*. Click **Next** to display the **Connection Information** dialog



Upgrade Wizard - Connection Information

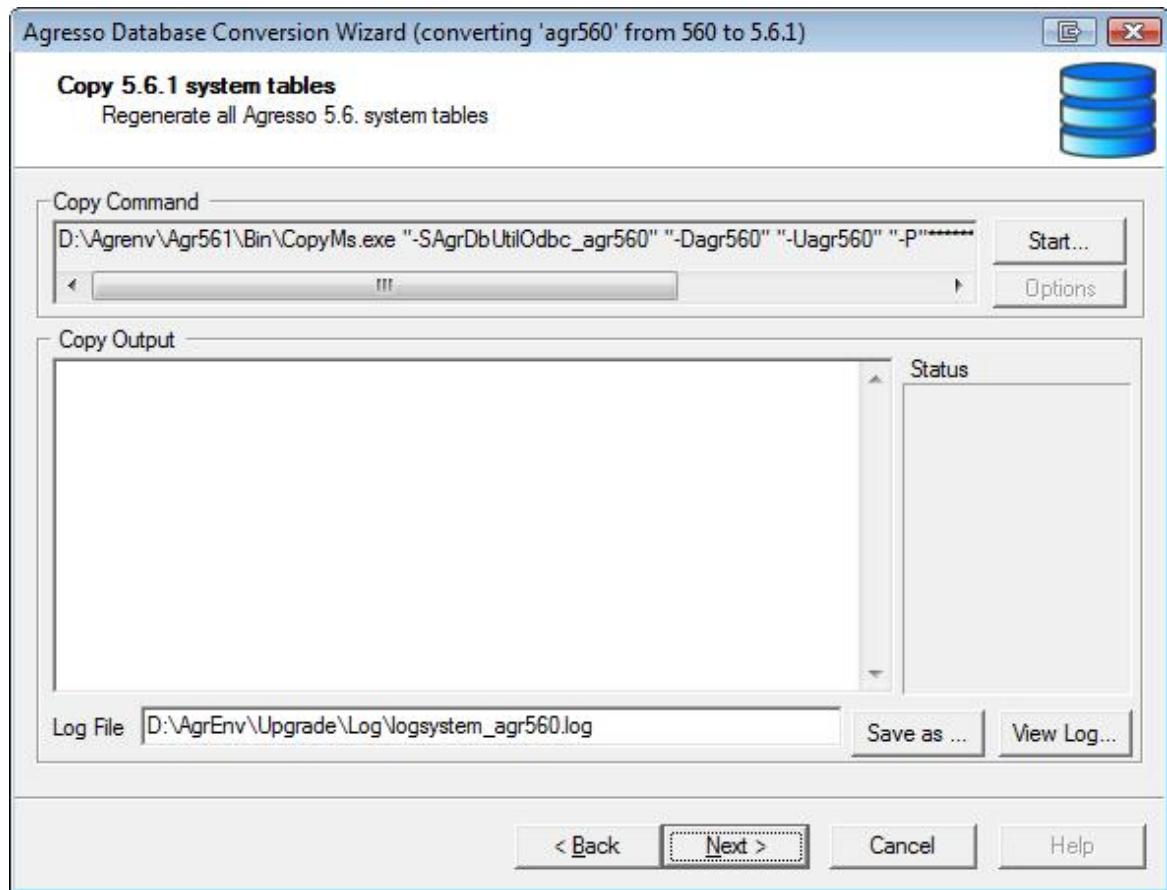


2. Select the data source, enter password, and click **Next**.

If all goes well, the wizard will display the conversion database name. Click **Next** to continue.

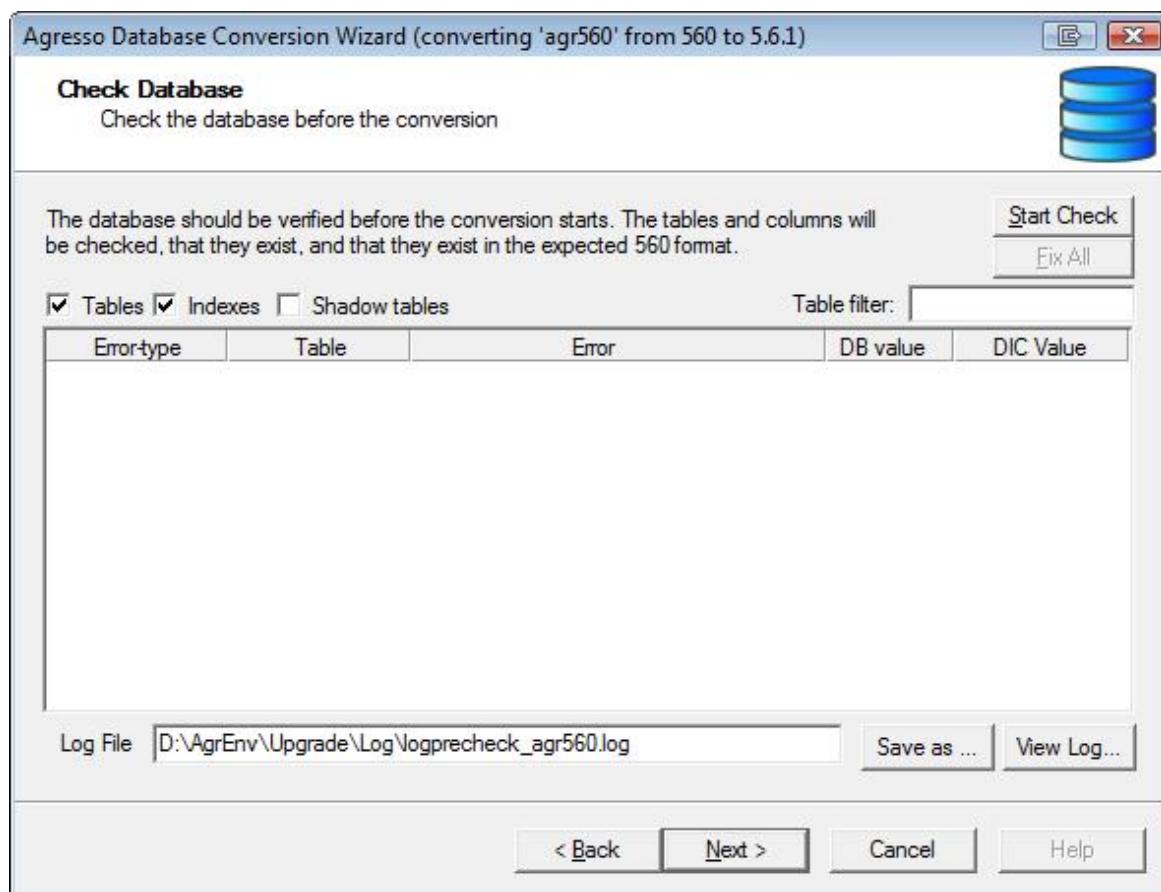
The wizard takes you to the **Copy 5.6 system tables** step.

[Upgrade Wizard - Copy in system tables](#)



3. Click **Start** to recreate the system and directory tables. When completed, click **Next** to continue with **Check Database**.

[Upgrade Wizard - Check Database](#)



This step verifies that the database is in the expected 5.6.0 format. This check might take some time.

4. Click **Start Check**

If you find differences, try the **Fix All** button. The Fix All might take some if tables with a lot of data needs to be fixed.

It is important to have a fixed database before continuing with the upgraded.

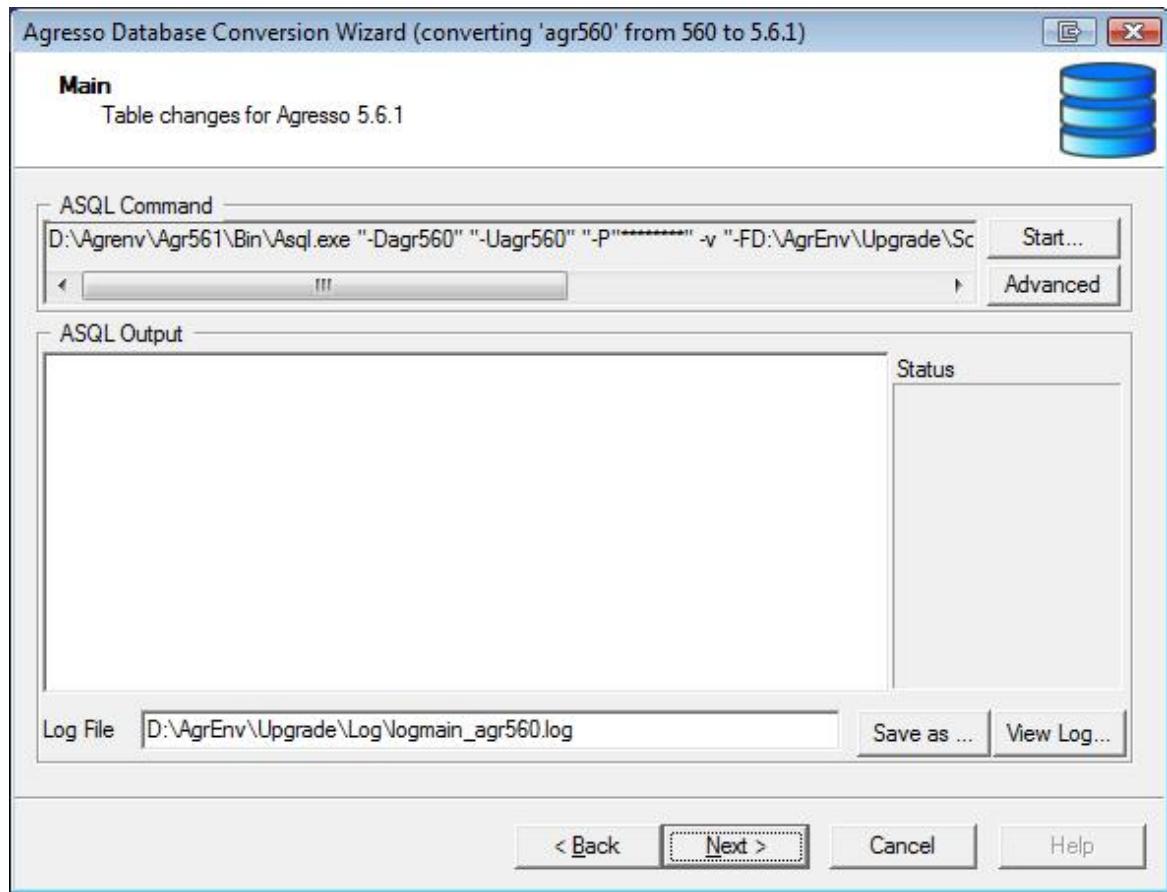
The following differences can be ignored:

- Columns in the table are not found in the dictionary.
- Columns are longer than expected.
- Errors on indexes (indexes will be recreated in a later step).

Click **Next** to continue.

You can now introduce the main changes in the database structure:

Upgrade Wizard – Main – Table changes for Agresso 5.6.1

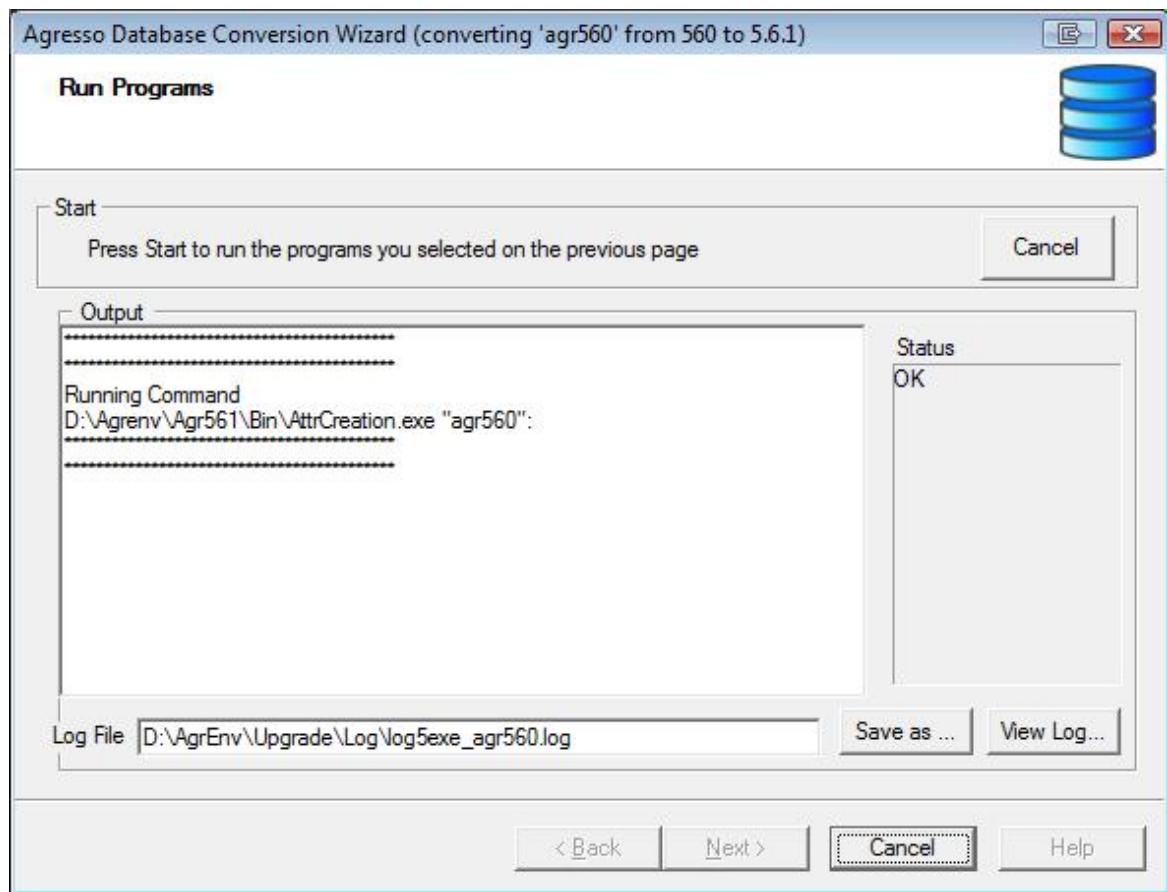


5. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

The next step is named **Select programs to run**. The available options depend on the version you are upgrading from.

As a rule, you should let the wizard make the selection for you, then click **Next**. This will take you to the **Run programs** step:

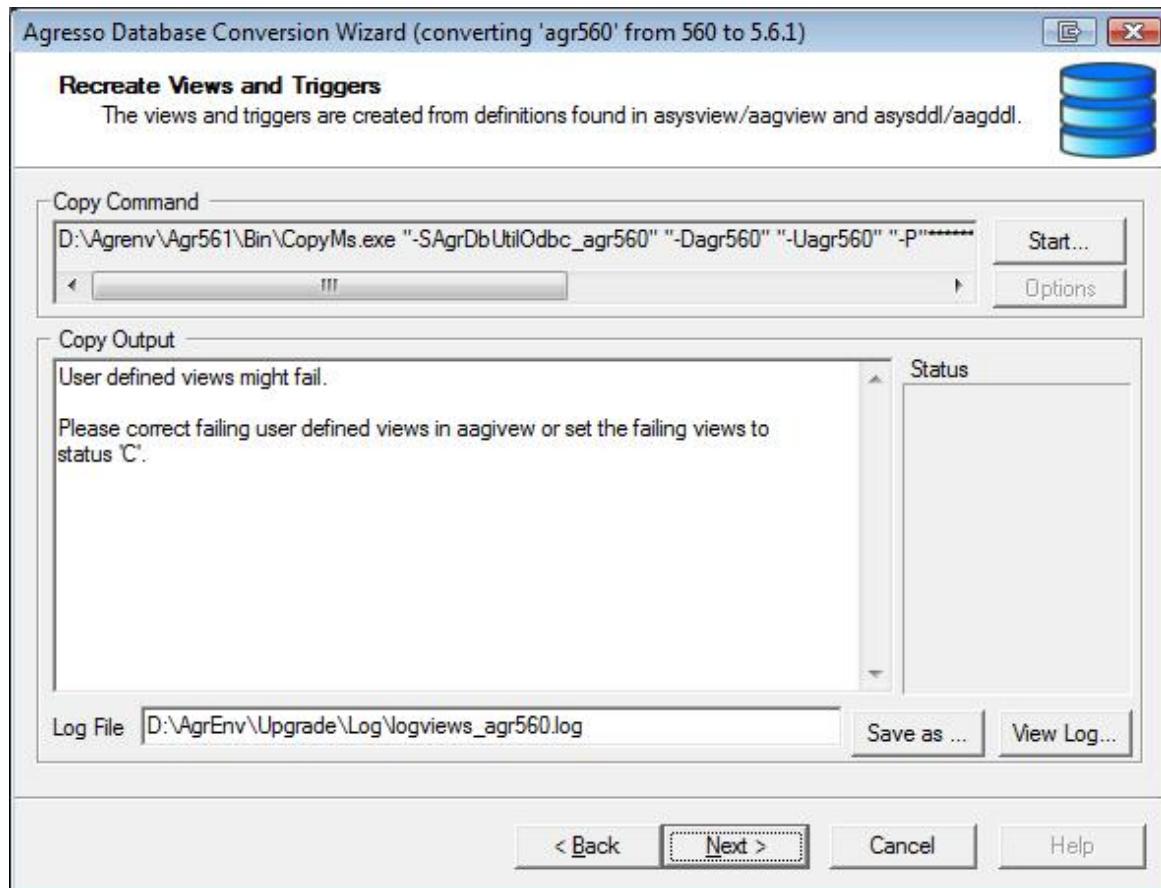
Upgrade Wizard - Run programs



6. Click **Start** to run the selected program, and then **Next** when the programs are completed.

This will take you to the **Recreate Views and Triggers** step.

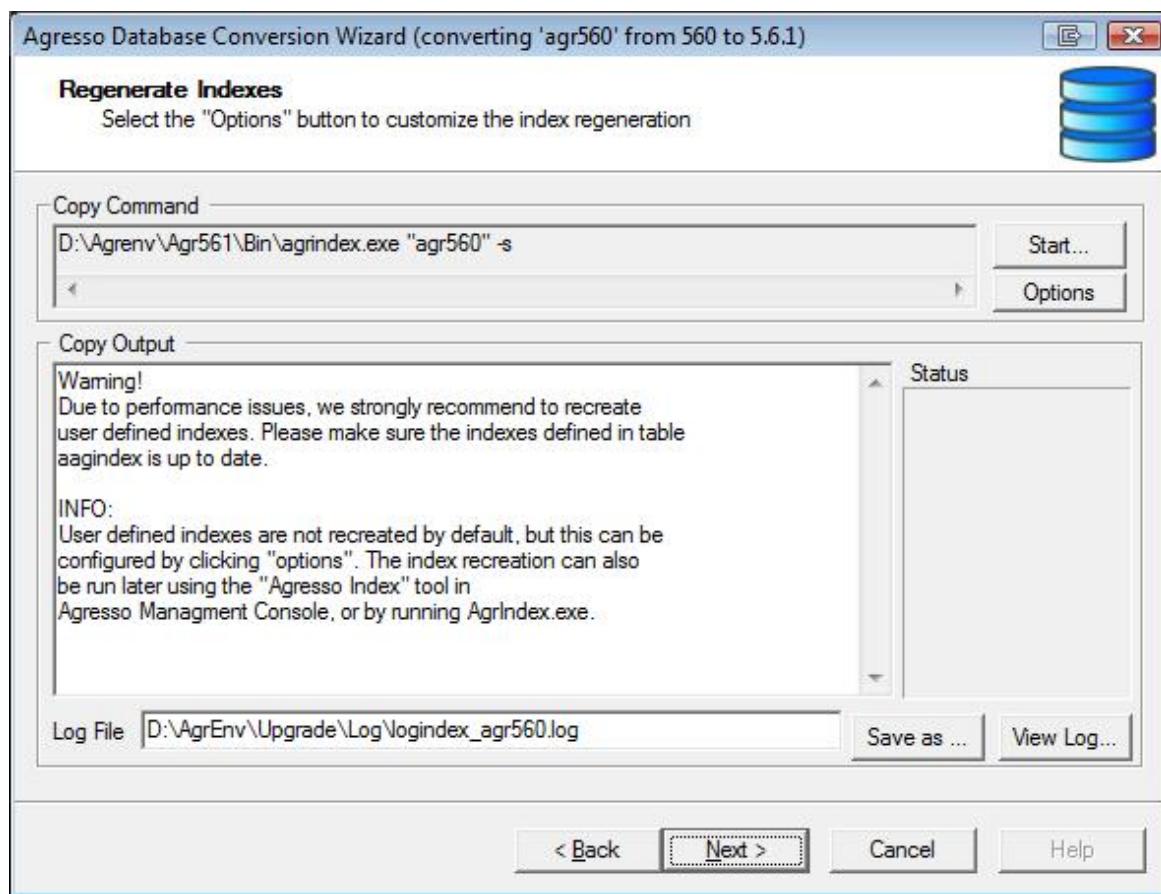
Upgrade Wizard - Recreate Views and Triggers



User defined views and triggers might fail if data structure has been changed in the latest release on the tables referred to. Definition of the views and triggers can be found in the tables aagview and aagddl. See in the Appendix for table/column changes.

7. Click **Start** and **Next** in the next windows, in order to restore system and user defined views and triggers. This will bring you to the final step.

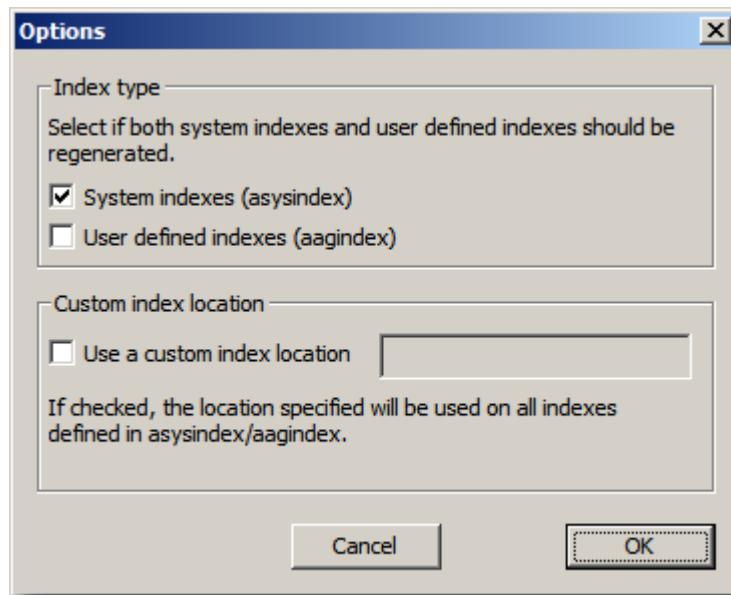
Upgrade Wizard - Regenerate Indexes



8. Click **Start** to regenerate the indexes. The indexes are important for performance and for preventing duplicates in the database.

Duplicates: If an index is not created due to duplicate rows, remove the duplicates, and rerun **Recreate indexes**.

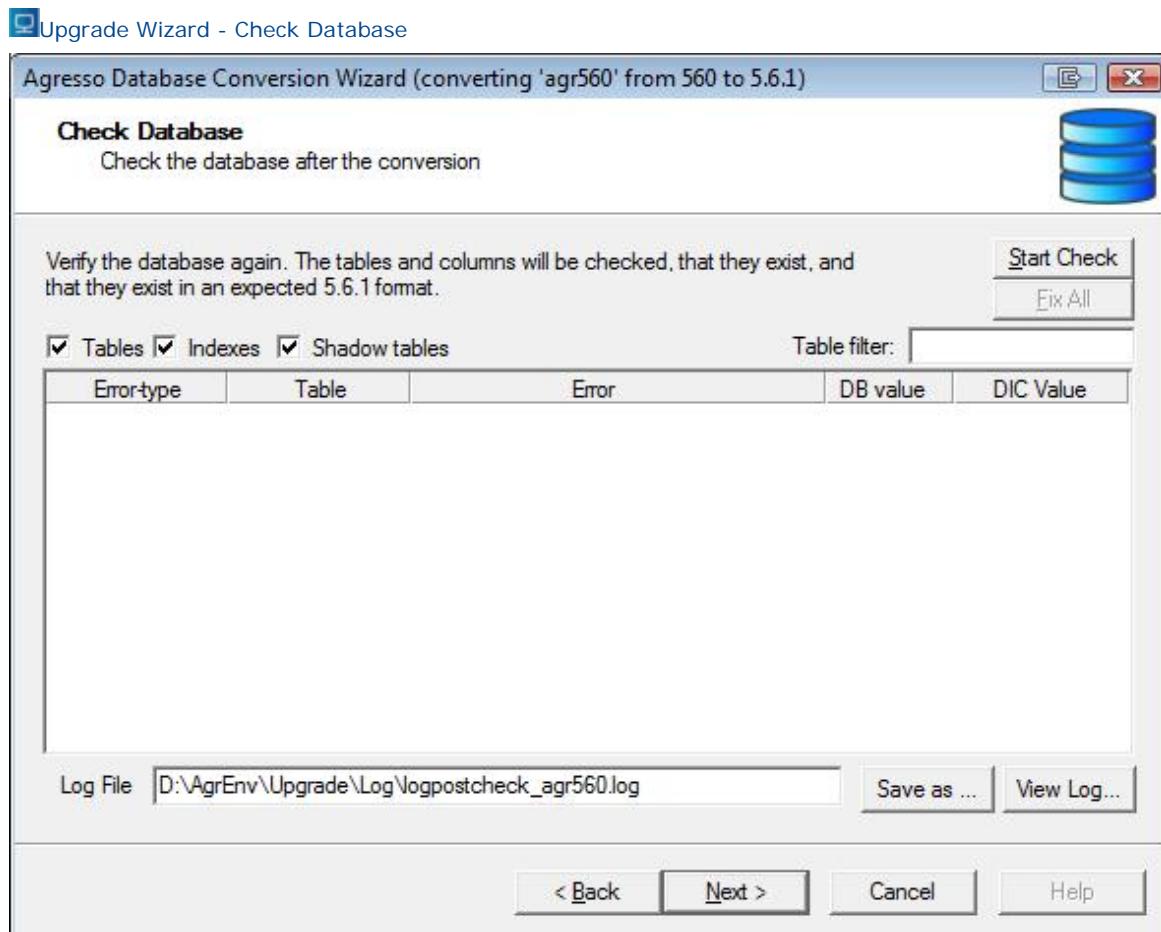
9. Click the **Options** button to open the Options dialog:



A note about indexes

By default, user defined indexes (stored in **aagindex**) are not recreated. Due to table changes, you should also change these, to avoid reduced performance. For details, see [AgrIndex](#).

10. Select your options and click **OK**. Back in the **Regerate indexes** dialog, click **Next**. You should now check the database after conversion.



Note: This check is ODBC dependent, and will create an ODBC for you if **Use custom ODBC Data Source** is not checked.

11. Click **Start Check**.

FINALISE UPGRADE

Overview

To complete the upgrade, you should:

- Check the converted Browser templates, and correct any errors.
- Clean up duplicates and add indexes.
- Correct all user defined views due to the table changes (see Appendix, Agresso Data Dictionary)
- Remove all tables no longer in use.

Check and correct Browser templates

Tools are provided to administer and ease the upgrade process for Browser templates:

- A Browser checker utility, `BrowserTemplateChecker.exe`, introduced in 5.5 Service Pack 1 that is run after upgrade.
- Check the log file produced during the upgrade. In this log, detailed information can be found of the Browser templates which need to be adjusted to run correctly on Agresso Business World 5.6.x. To fix the problem, open the browser template,

make the necessary changes and save.

Indexes and duplicates

When running the database check step in the wizard, there might be indexes missing due to duplicates.

Indexes might be very important, and missing or wrong indexes can lead to very poor and slow performance.

Check the log files if there have been any errors while creating the indexes.

See log file ..Database Script\Upgrade\Log\logindex_<database>.log

Use any database tool to find the duplicates (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

Example on how to find the duplicates:

```
select distinct client, attribute_id from agldimension
group by client, attribute_id
having count(*) > 1
order by client, attribute_id
/
```

Delete/change so there are no duplicates in the table, and re-create the index.

Look up in the Appendix, Agresso Data Dictionary to see the correct index definition. Create the index using the preferred DBA Tool (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

User defined views

The table structure is changed from release to release. Ensure all user defined objects are changed according to the new structure (See Appendix Agresso Data Dictionary)

Change the definition and recreate the views

See log file ..Database Script\Upgrade\Log\logviews_<database>.log

Remove tables not longer in use

The script *drop.asp* in the *Scripts* directory will drop all old and temporary tables no longer used by the application.

Note: This script must be run when the upgrade has been completely verified, and you are sure that none of the old tables are needed for backup purposes.

CREATE SYSTEM ATTRIBUTES

Description

The wizard **Create System Attributes** will generate all the fixed attributes currently missing in your ABW 5.6.1 installation.

The wizard is part of Step 6 – **Run binary programs** in the main system upgrade wizard, but is also available in a stand alone version, available in the file *CreateAttributesWizard.exe*.

Standard (fixed) attributes only

The wizard updates only attributes delivered as part of the standard **Agresso Business World** installation (with attribute id from A0 to LZ).

Other attribute types, for example localisation specific attributes or custom attributes, will not be updated by the wizard.

All clients will be updated

Attributes are client dependent and the wizard will generate one attribute instance per client. Attribute names and descriptions will comply with the selected language (Company Information screen, CR01) for the various clients.

Run the wizard

The wizard is very straightforward, with no complex options. Just follow the instructions on screen.

From Agresso 5.5 SPx to Agresso 5.6

NOTES ON UPGRADING FROM 5.5.X TO 5.6.1

Prerequisites

- The new version of **Agresso Business World** must have been successfully installed before you start the upgrade process.
- We assume that you have profound knowledge of the Agresso installation and the RDBMS used.

Prepare the upgrade

Backup the database

Before you run the upgrade wizard, you should back up your Agresso data. To be completely safe, you should also use the **Agresso Copy** program to make a complete copy of all database tables.

Verify disk space

Available disk space needed for the upgrade is approximately the size of the largest table + 10%.

See the list of table changes in the Appendix.

Create 5.6 Data Source and initialize Business Server environment

Use the **Agresso Management Console** (AMC) to create a new 5.6 Data Source connected to the old database.

When the connection is up and working, you must also initialise the Business Server environment (select the **Business server** node in **AMC**) and then **Initialise Business Server.**).

Now you can run the upgrade wizard. See [Upgrading Procedure](#).

amendment tables

Logging

If amendment logging is turned on for an Agresso table, an amendment table (or shadow table) are created and continuously updated with all table changes.

Note: During upgrade, all amendment logging will be turned off (by the wizard), and switched on again when the convert script is complete.

Table for amendment tables

All amendment tables are defined in the table [aagamendlog](#).

Naming standard: An Agresso table name is constructed from the structure `a<module><identification>`, while an amendment table is extended with the letters `shd` between `<module>` and `<identification>`.

If you turn on amendment logging for `acrclient`, the amendment table `acrshdclient` will be created and added to `aagamendlog`.

Recreate active logging triggers

If some of the shadow tables are not converted correctly, and the upgrade wizard is unable to fix the error in the database check. The shadow tables can be re-created by renaming or dropping the existing table, and then re-enable amendment logging for the table in the [Activation of logging server](#) (AG30) page in the smart client.

When errors occur

It is important that all errors encountered during the upgrade process are reported back to Agresso.

Customer

As a customer you are asked to report the errors as tickets in the Heat system.

Partner or subsidiary

If you, as a partner or subsidiary, have access to the Heat system, report the errors as Heat tickets. Otherwise, please report the errors to the Technical Forum - Agresso 5.6 Technical Consultants <http://forum.agresso.no/>.

Do not report the problem if it is related to duplicates in the database, full disk, full tablespace or similar.

Check user defined queries after upgrading

As part of the system upgrade, the ASQL parser has been improved with stronger syntax check and more consistent error handling. If a complex query contains an error, no part of the query will be executed. Instead, ASQL will report a parser error.

Previously, valid ASQL appearing before the error would have been executed, while the rest would be ignored. Now, the query will not run at all.

It is therefore important that you run all user defined queries after the upgrade.

UPGRADING PROCEDURE

The upgrade wizard

You will use the wizard located in

...Agresso 5.6\DatabaseScript\DbUpgrade\UpgradeWizards.exe.

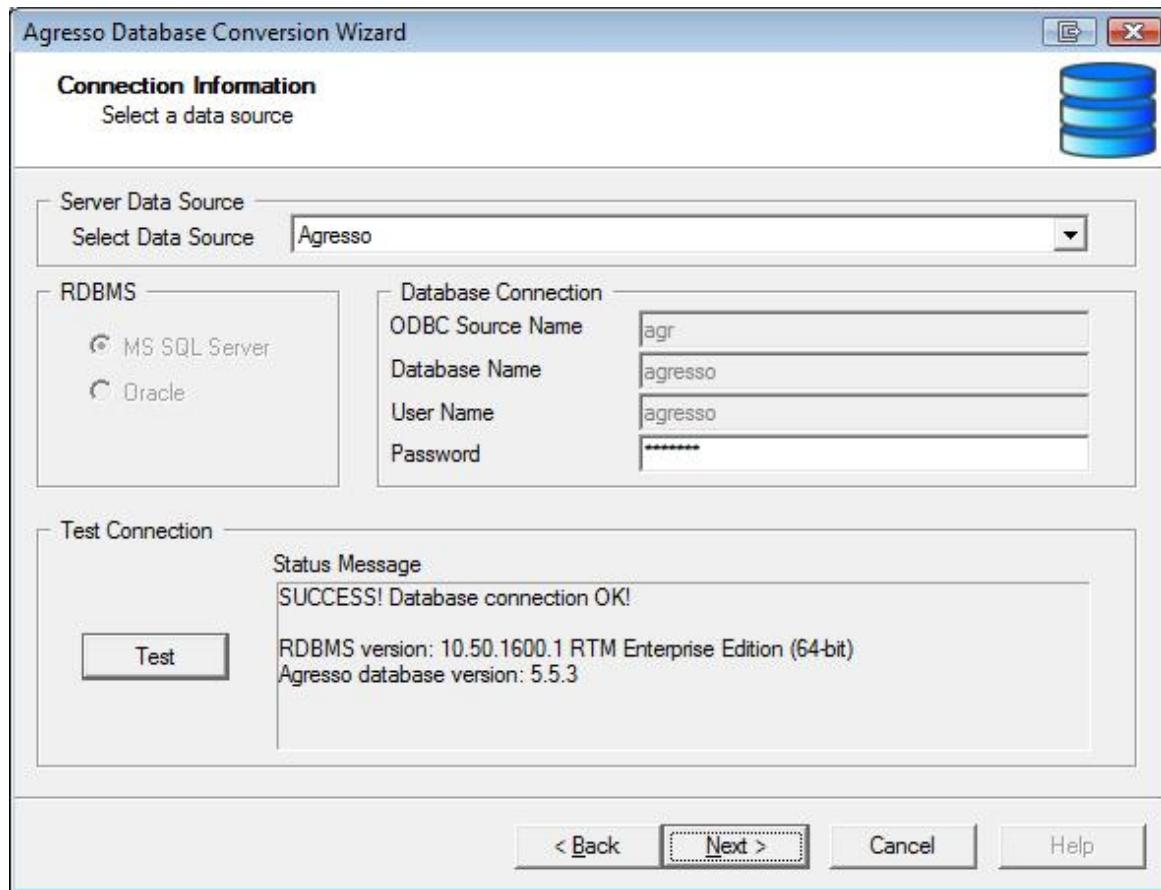
Start the Wizard

- A. Start [UpgradeWizards.exe](http://...Agresso 5.6\DatabaseScript\DbUpgrade\UpgradeWizards.exe)
- B. Select **Convert from 5.5**

Procedure

1. When the upgrade wizard is up and running, select Step 1, [Main Upgrade Wizard](#). Click **Next** to display the **Connection Information** dialog





2. Select the data source, enter password, and click **Next**.

If all goes well, the wizard will display the conversion database name. Click **Next** to go to the **Window Options** step.

Note: the **Window Options** step is only relevant if you convert from 5.5.2.

2A: Do you convert from 5.5.2?

If YES:

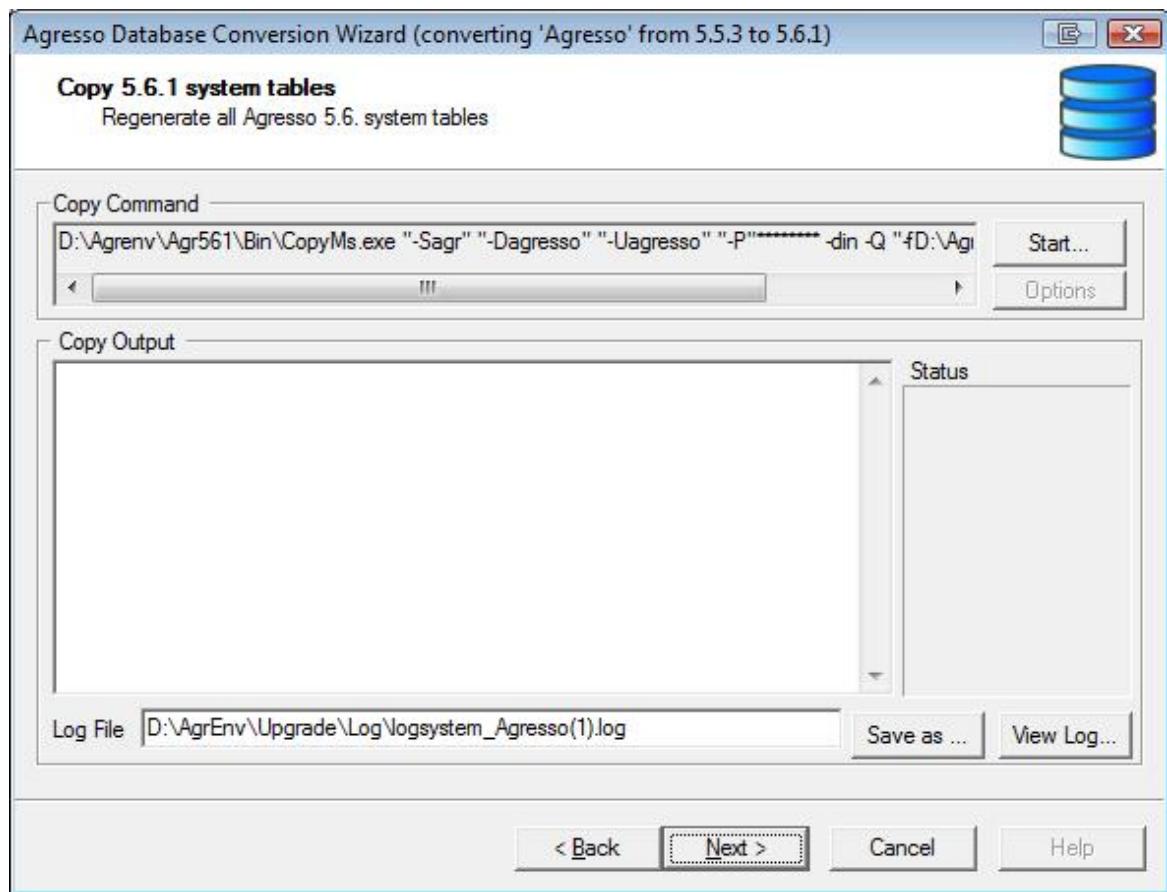
- Click **Start** in the **Window Options** dialog and wait for the wizard to complete.
- Click **Next** to continue.

If NO:

- Click **Next** to continue.

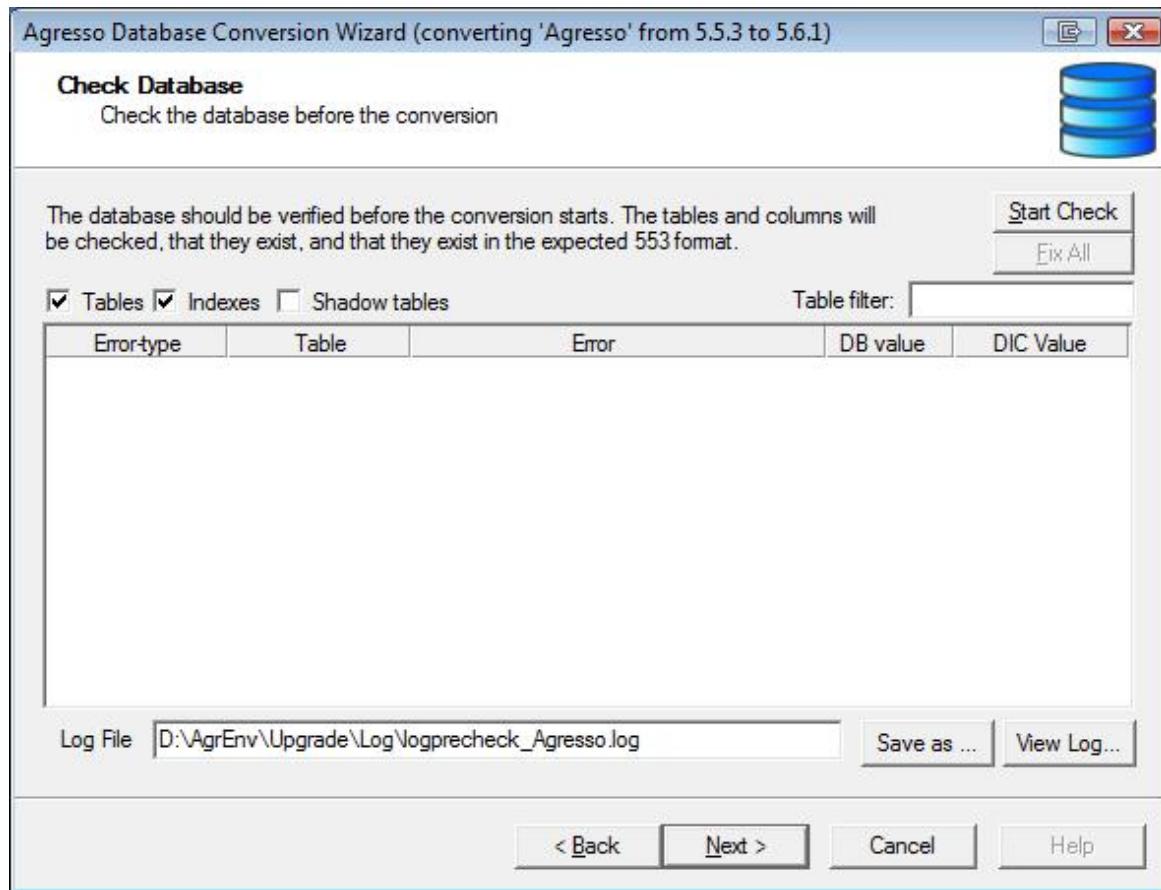
The wizard takes you to the **Copy 5.6 system tables** step.

Upgrade Wizard - Copy in system tables



3. Click **Start** to recreate the system and directory tables. When completed, click **Next** to continue with **Check Database**.

[Upgrade Wizard - Check Database](#)



This step verifies that the database is in the expected 5.5 format. This check might take some time.

4. Click **Start Check**

If you find differences, try the **Fix All** button. The Fix All might take some if tables with a lot of data needs to be fixed.

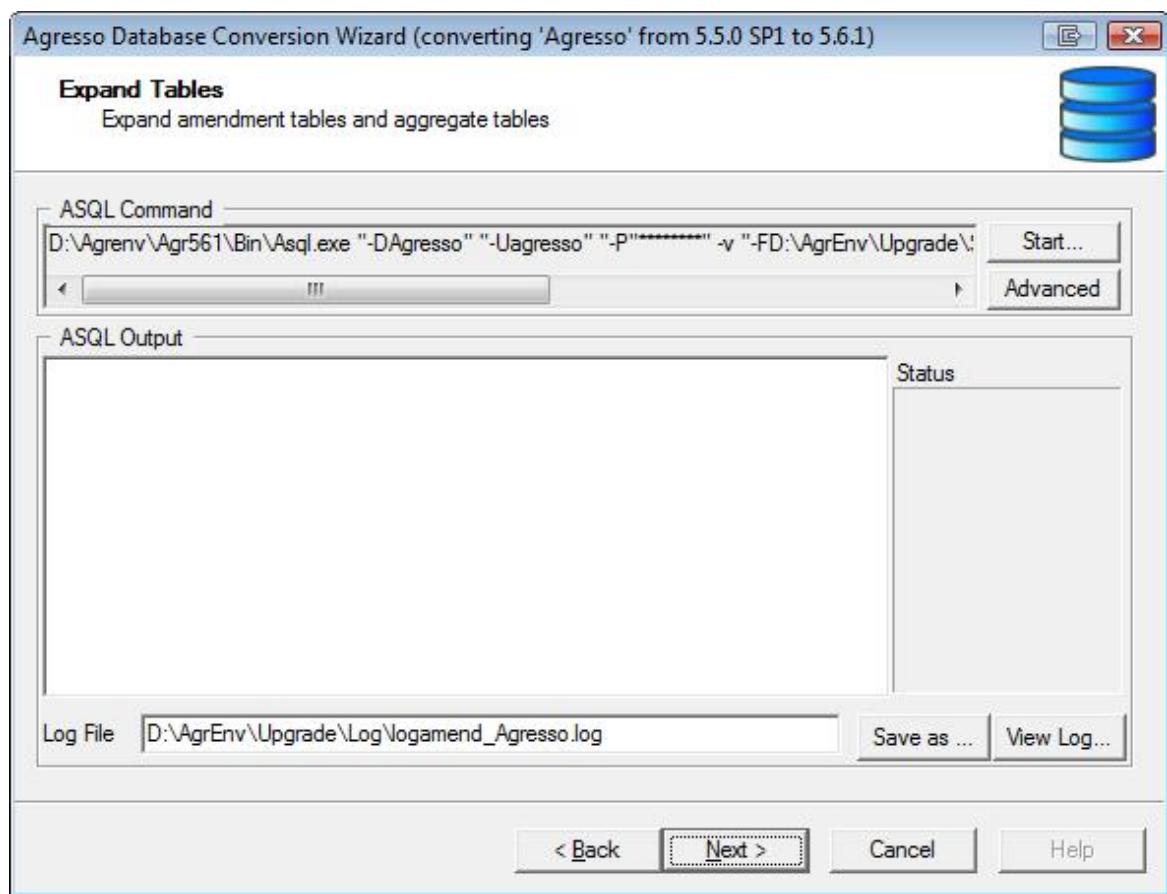
It is important to have a fixed database before continuing with the upgraded.

The following differences can be ignored:

- Columns in the table are not found in the dictionary.
- Columns are longer than expected.
- Errors on indexes (indexes will be recreated in a later step).

The next step is **Expand Tables**.

 **Upgrade Wizard - Expand Tables** (from 5.5 SP1 only)



4. A Do you convert from 5.5.1?

If YES:

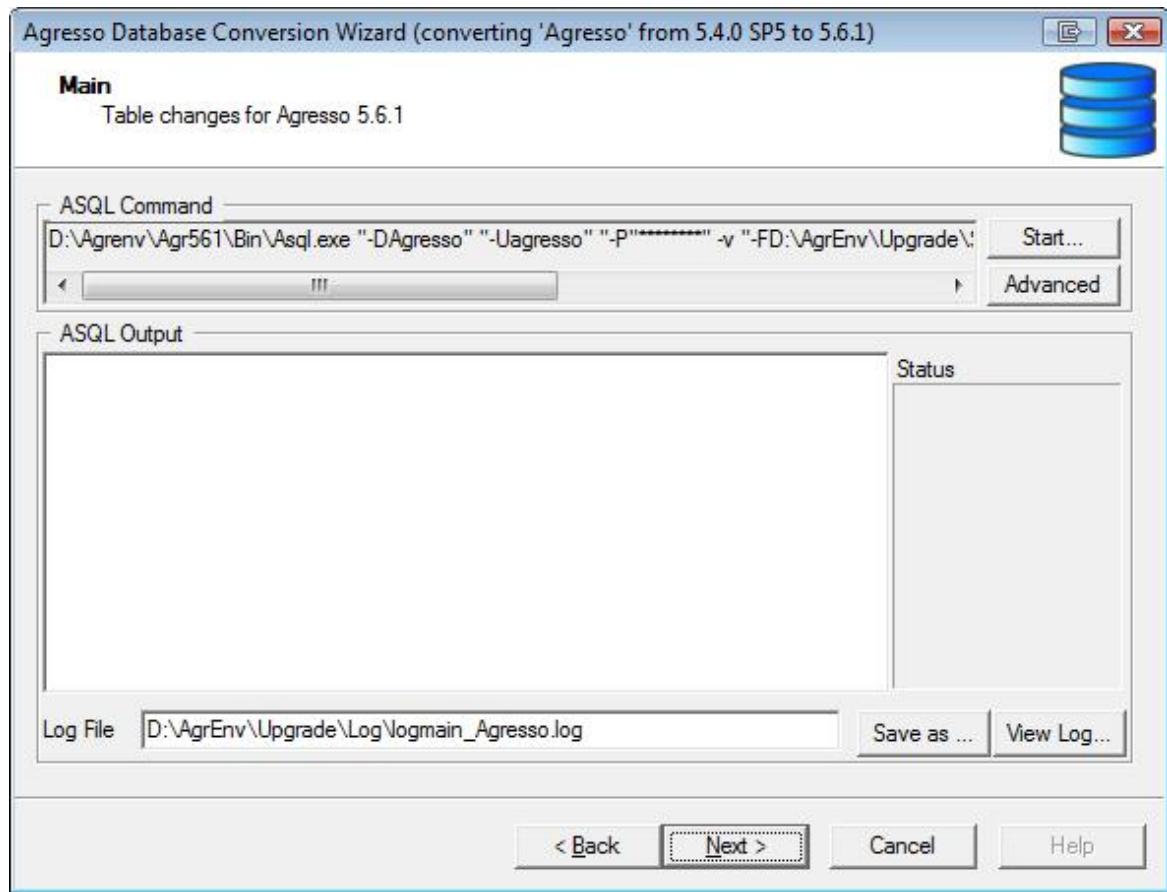
- Click **Start** in the **Expand Tables** dialog and wait for the wizard to complete. This step expand columns in the amendment tables and the aggregate tables. It may take some time.
- Click **Next** to continue.

If NO:

- Click **Next** to continue.

You can now introduce the main changes in the database structure:

[Upgrade Wizard – Main – Table changes for Agresso](#)

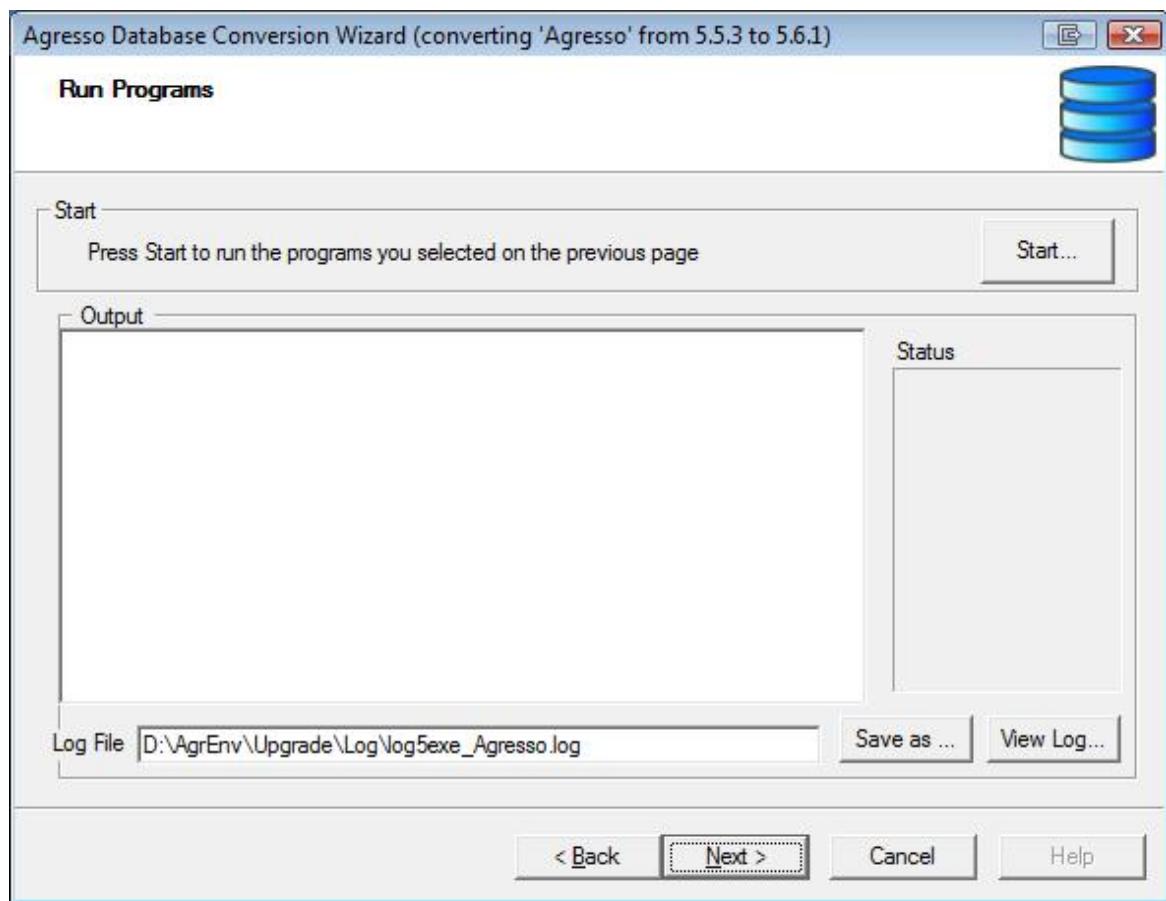


5. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

The next step is named **Select programs to run**. The available options depend on the version you are upgrading from.

As a rule, you should let the wizard make the selection for you, then click **Next**. This will take you to the **Run programs** step:

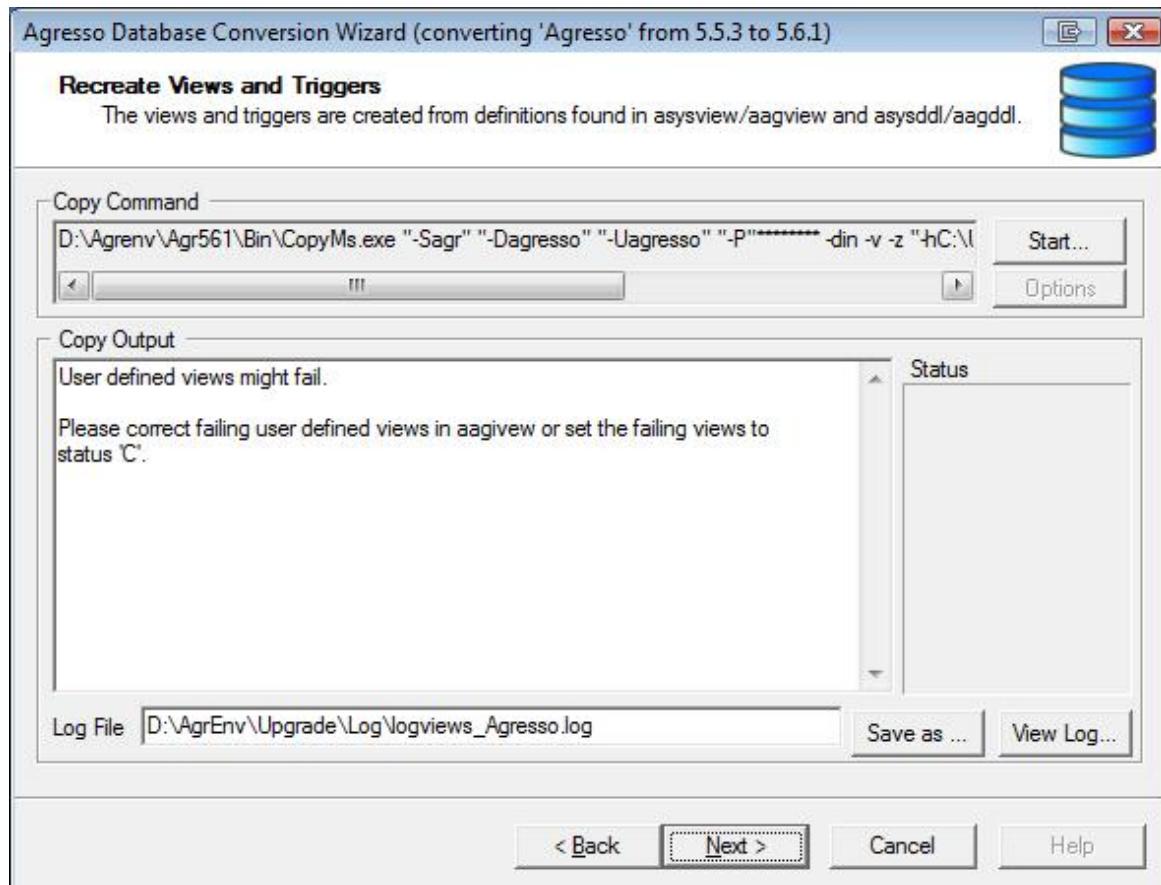
Upgrade Wizard - Run programs



6. Click **Start** to run the selected program, and then **Next** when the programs are completed.

This will take you to the **Recreate Views and Triggers** step.

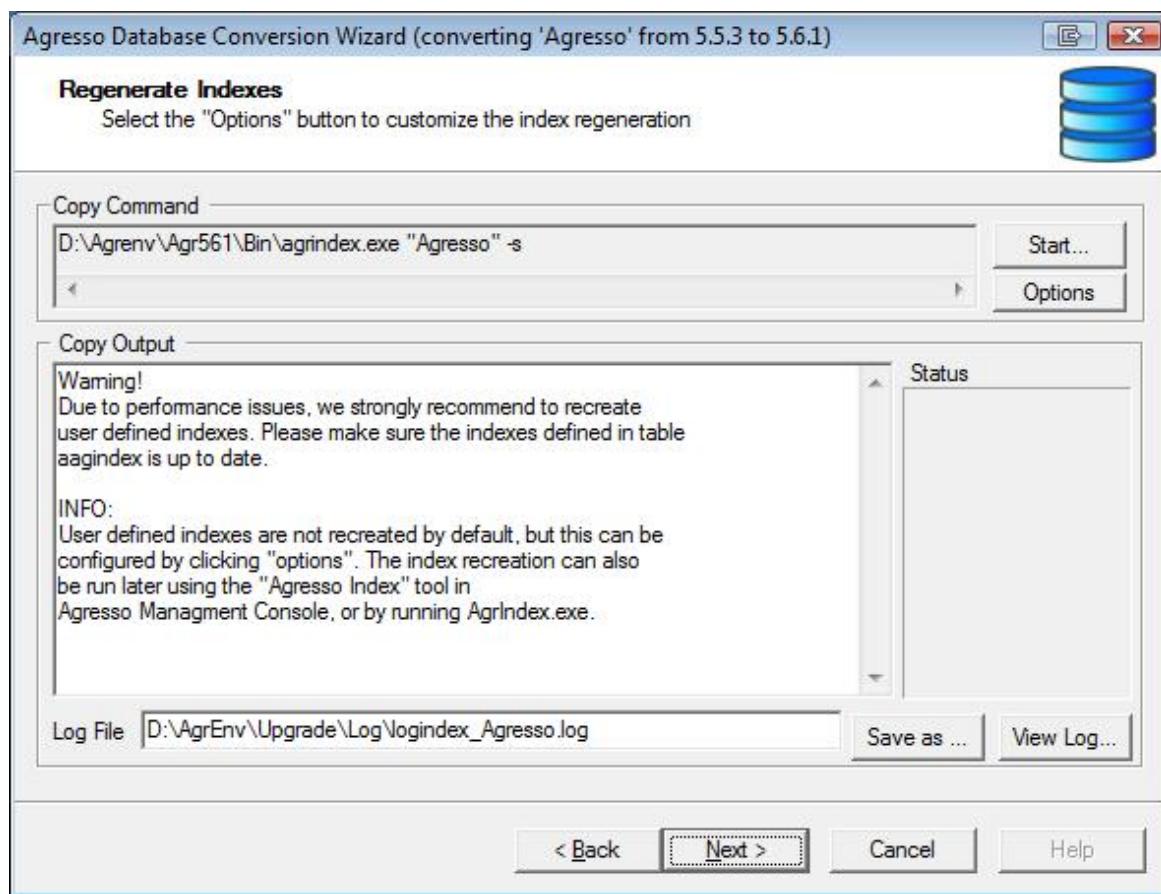
[Upgrade Wizard - Recreate Views and Triggers](#)



User defined views and triggers might fail if data structure has been changed in the latest release on the tables referred to. Definition of the views and triggers can be found in the tables aagview and aagddl. See in the Appendix for table/column changes.

7. Click **Next** and **Start** in the next windows, in order to restore system and user defined views and triggers. This will bring you to the final step.

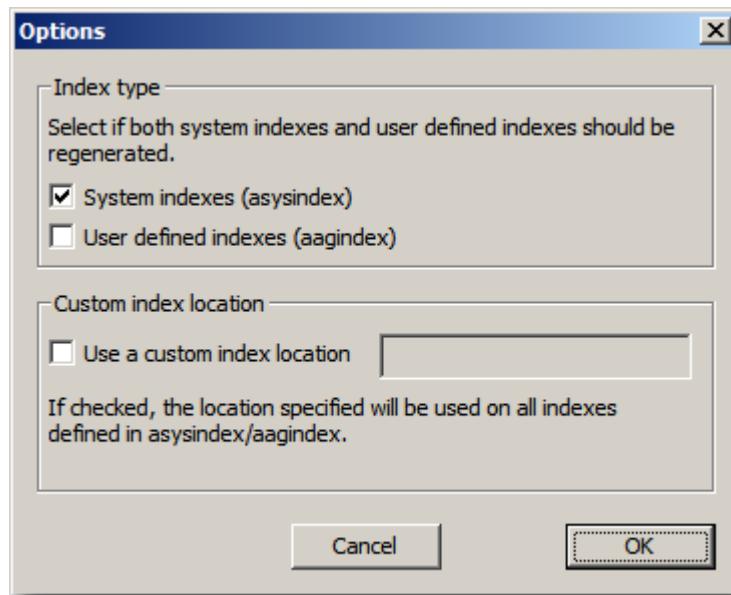
Upgrade Wizard - Regenerate Indexes



8. Click **Start** to regenerate the indexes. The indexes are important for performance and for preventing duplicates in the database.

Duplicates: If an index is not created due to duplicate rows, remove the duplicates, and rerun **Recreate indexes**.

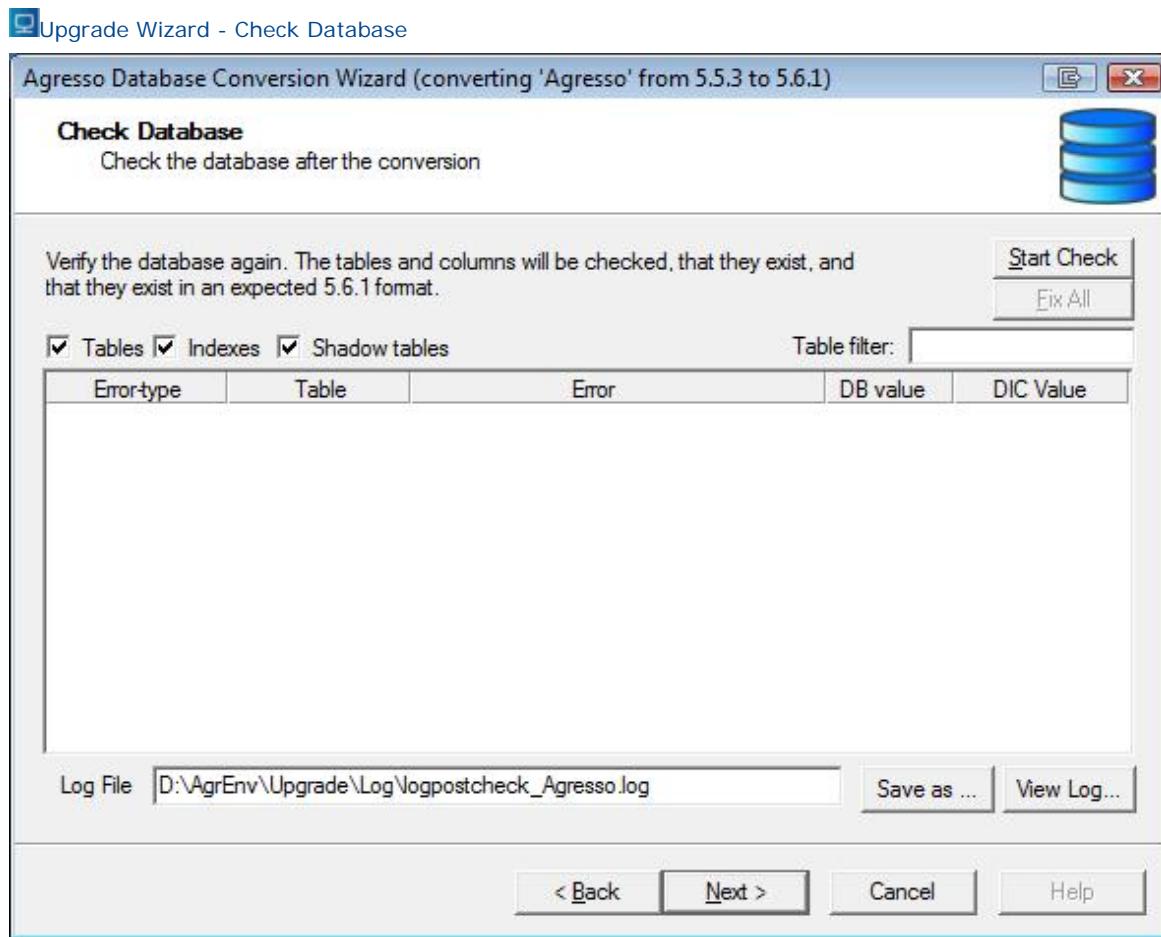
9. Click the **Options** button to open the Options dialog:



A note about indexes

By default, user defined indexes (stored in **aagindex**) are not recreated. Due to table changes, you should also change these, to avoid reduced performance. For details, see [AgrIndex](#).

10. Select your options and click **OK**. Back in the **Regerate indexes** dialog, click **Next**. You should now check the database after conversion.



11. Click **Start Check**.

The database upgrade to 5.6 is now finished!

FINALISE UPGRADE

Overview

To complete the upgrade, you should:

- Check the converted Browser templates, and correct any errors.
- Clean up duplicates and add indexes.
- Correct all user defined views due to the table changes (see Appendix, Agresso Data Dictionary)
- Remove all tables no longer in use.

Check and correct Browser templates

Tools are provided to administer and ease the upgrade process for Browser templates:

- A Browser checker utility, `BrowserTemplateChecker.exe`, introduced in 5.5 Service Pack 1 that is run after upgrade.

- Check the log file produced during the upgrade. In this log, detailed information can be found of the Browser templates which need to be adjusted to run correctly on Agresso Business World 5.6.x. To fix the problem, open the browser template, make the necessary changes and save.

Indexes and duplicates

When running the database check step in the wizard, there might were indexes missing due to duplicates. Indexes might be very important, and missing or wrong indexes can lead to very poor and slow performance.

Check the log files if there has been any errors while creating the indexes.

See log file ..Database Script\Upgrade\Log\logindex_<database>.log

Use any database tool to find the duplicates (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

Example on how to find the duplicates:

```
select distinct client, attribute_id from agldimension
group by client, attribute_id
having count(*) > 1
order by client, attribute_id
/
```

Delete/change so there are no duplicates in the table, and re-create the index.

Look up in the Appendix, Agresso Data Dictionary to see the correct index definition. Create the index using the preferred DBA Tool (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

User defined views

The table structure is changed from release to release. Ensure all user defined objects are changed according to the new structure (See Appendix Agresso Data Dictionary)

Change the definition and recreate the views

See log file ..Database Script\Upgrade\Log\logviews_<database>.log

Remove tables not longer in use

The script *drop.asp* in the *Scripts* directory will drop all old and temporary tables no more used by the application.

Note: This script must be run when the upgrade has been completely verified, and you are sure that none of the old tables are needed for backup purposes.

CREATE SYSTEM ATTRIBUTES

Description

The wizard **Create System Attributes** will generate all the fixed attributes currently missing in your ABW 5.6 installation.

The wizard is part of Step 6 – **Run binary programs** in the main system upgrade wizard, but is also available in a stand alone version, available in the file *CreateAttributesWizard.exe*.

Standard (fixed) attributes only

The wizard updates only attributes delivered as part of the standard **Agresso Business World** installation (with attribute id from A0 to LZ).

Other attribute types, for example localisation specific attributes or custom attributes, will not be updated by the wizard.

All clients will be updated

Attributes are client dependent and the wizard will generate one attribute instance per client. Attribute names and descriptions will comply with the selected language (Company Information screen, CR01) for the various clients.

Run the wizard

The wizard is very straightforward, with no complex options. Just follow the instructions on screen.

From Agresso 5.4 to Agresso 5.6

INTRODUCTION TO UPGRADING

Important changes

Changes in database

The Agresso upgrade from version 5.4 to 5.6 affects most of the Agresso tables. Many tables have new columns, and, to support more intuitive field values, several (old) column definitions have been modified. For example:

- Transaction series are expanded to 12 digits. Bigint(MSSql) and number(20) (Oracle).
- Account numbers are expanded to 25 characters.
- All attribute values can now hold up to 25 characters.
- Descriptions, names, file_names etc. are expanded to 255 characters.
- All supplier_id's and customer_id's (apar_id's) are converted to alphanumeric, 12 characters.
- Where client exists in the index, we have moved the client column to be listed first in the index list in order to improve the performance on index access.

Note: Although a large amount of columns have been redefined to allow extended values, this will not necessarily (or: immediately) affect the database size. The columns have been redefined from datatype **char** to **varchar**, allowing a more dynamic utilisation of the disk space.

Functional changes

Some functional areas have been completely rewritten, as a means to keep up with both functional and technological requirements. Affected areas are, for example:

- Document archive - which now can be utilised from all over Agresso.
- Authorisation and access rights - where the introduction of a new Role concept has required a new data model and implementation.
- Workflow - a new Agresso workflow engine has replaced the previous implementations (Compello).

Database updates

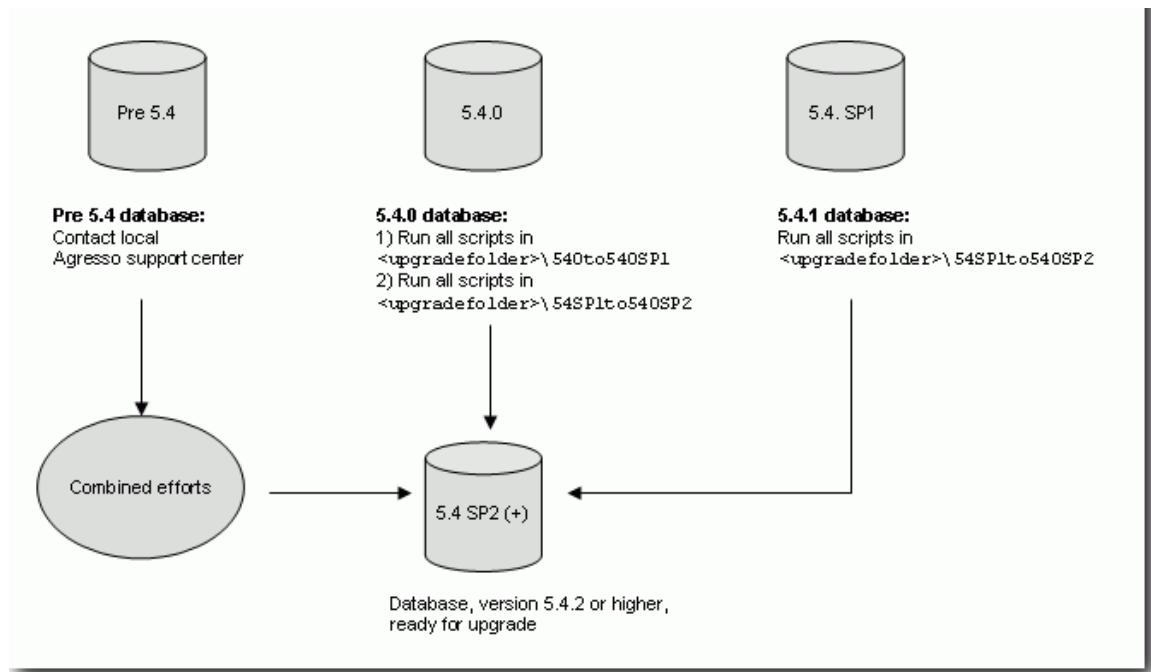
Please note:

- If your database is older than Agresso version 5.4 SP2, you must contact your local Agresso support center. In that case, you need assistance to make an additional, basic upgrade.

- The new version of Agresso Business World must have been successfully installed before you start the upgrade process.
- We assume that you have profound knowledge of the Agresso installation and the RDBMS used.
- **Upgrade scripts and wizards:** The upgrade scripts and wizards are found in the folder `<...Agresso 5.6>\DatabaseScript\DbType` on the upgrade DVD. It will be referred to as `<upgradefolder>`.

Diagram

The required database updates are shown in the following diagram:



Find correct database version

If you need to find the correct database version, execute the following sql statement:

```
select text1,text2,text3 from syssetup where name = 'BASE_VERSION'
```

The main upgrade stages

There are four main upgrade stages:

1. Installation of the new 5.6 software, database upgrade and data conversion. We refer to this stage as the Main system upgrade.
2. Additional system areas upgrades - dependent on the areas included in your license.
3. Functional areas (module) upgrades - dependent on modules used by your installation.
4. Completion.

Test versus Production

In the upgrade process, we make a distinction between upgrade of a *Test system* and upgrade of the *Production system*. For both systems, however, you must go through all the 4 stages mentioned above.

Test

The Test upgrade comes first, and results in a complete, fully functional 5.6 system. You will run through the complete upgrade process, and perform all tasks: run the upgrade wizards, and use the **Agresso Management Console** and the **ABW Smart Client** to set up and configure elements not handled by the wizards. When the Test system is found in order, you are ready for the Production update.

Production

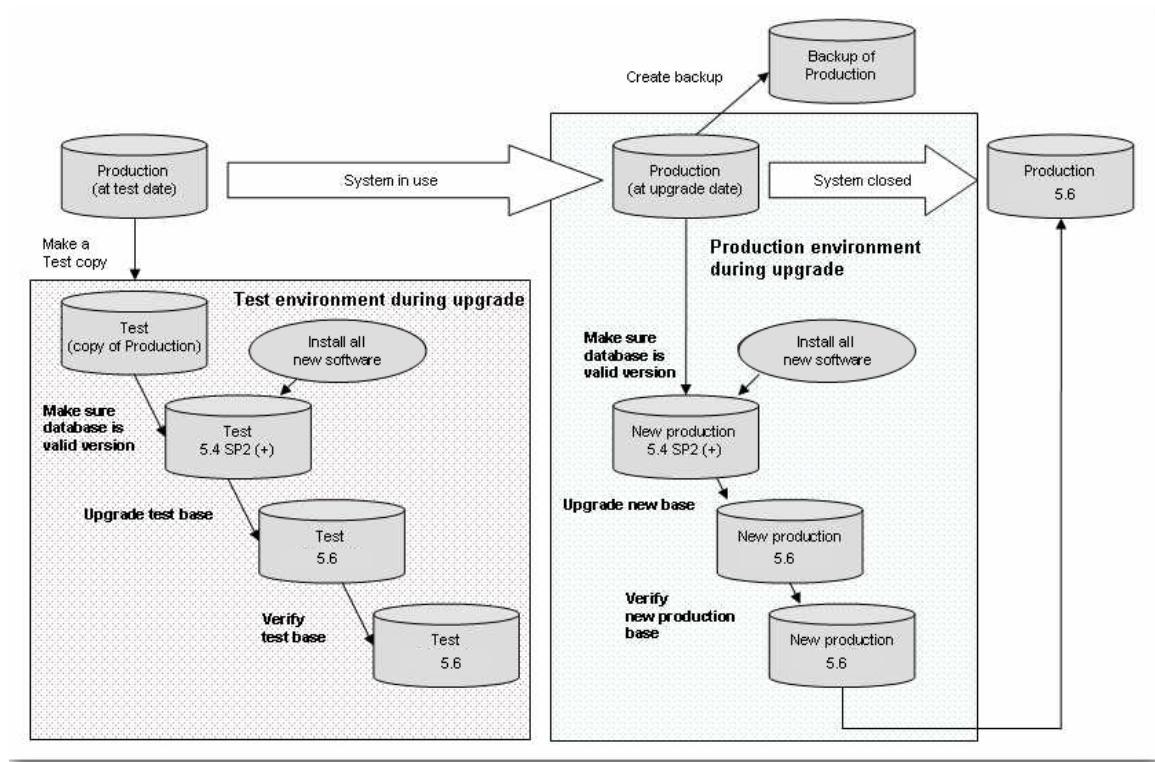
The Production upgrade is basically a repetition of the Test upgrade process, with one important difference: Most of the manual setup for the *additional system areas* in the Test environment, can be copied directly

into the Production database. The wizard **Copy Setup Between Databases** will facilitate the production upgrade process.

Note: If you do not complete a Test upgrade first, but works directly with the Production system, you will have to follow the detailed upgrade process as described under *Test System Upgrade (all tasks)*. With one system only, you will not have any previous settings to copy.

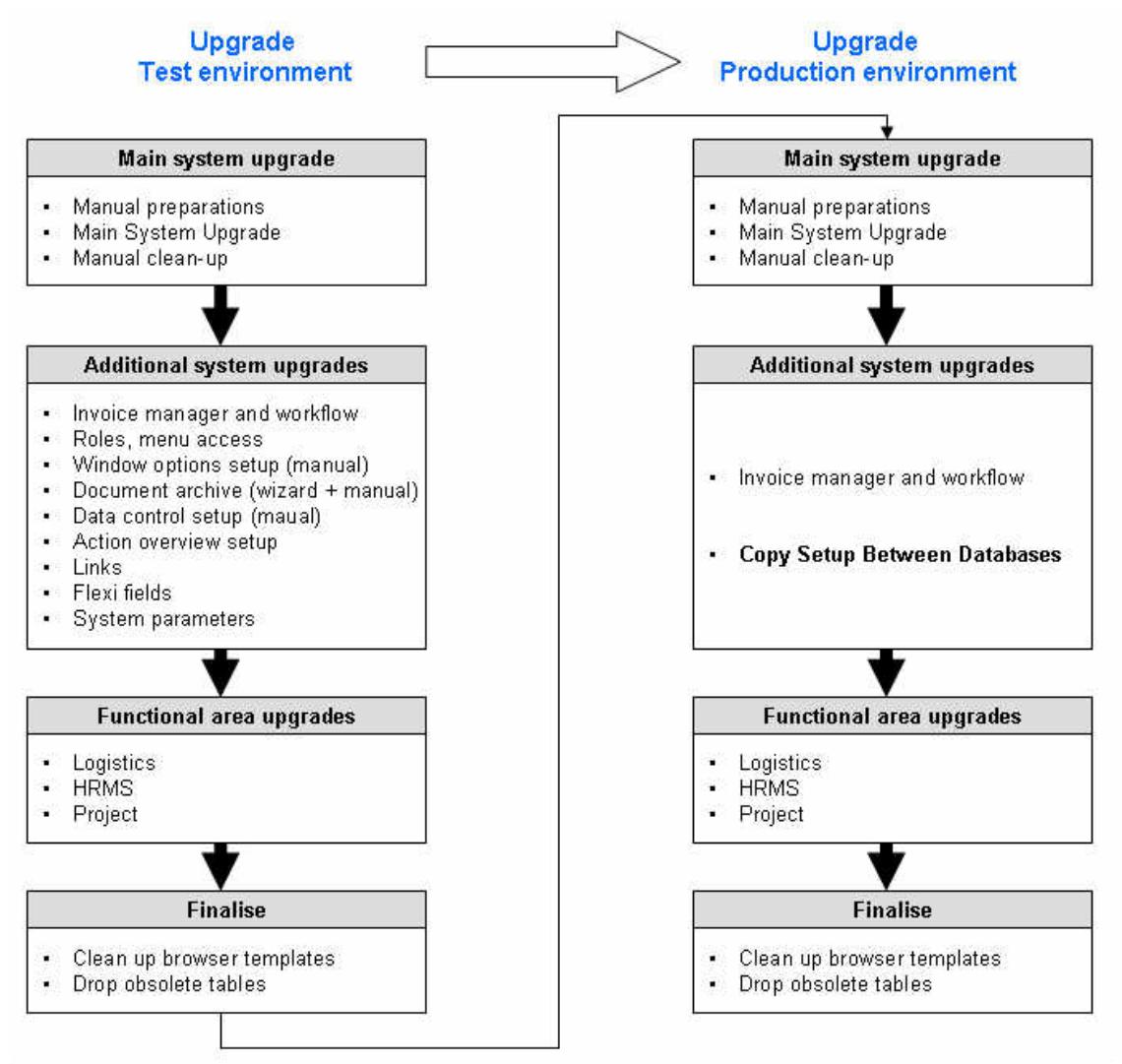
Upgrade overview diagram

The diagram below outline the complete upgrade process. The time aspect is indicated by showing the activities from left to right. There should be considerably less time involved in the final upgrade of the production database, compared to the test upgrade.



Test and production tasks

When we reduce the scope to the upgrade of a 5.4 SP2(+) system, the main tasks are as follows:



Wizards

All upgrade processes are wizard based. For some areas, however, it will also be necessary to perform additional, manual actions.

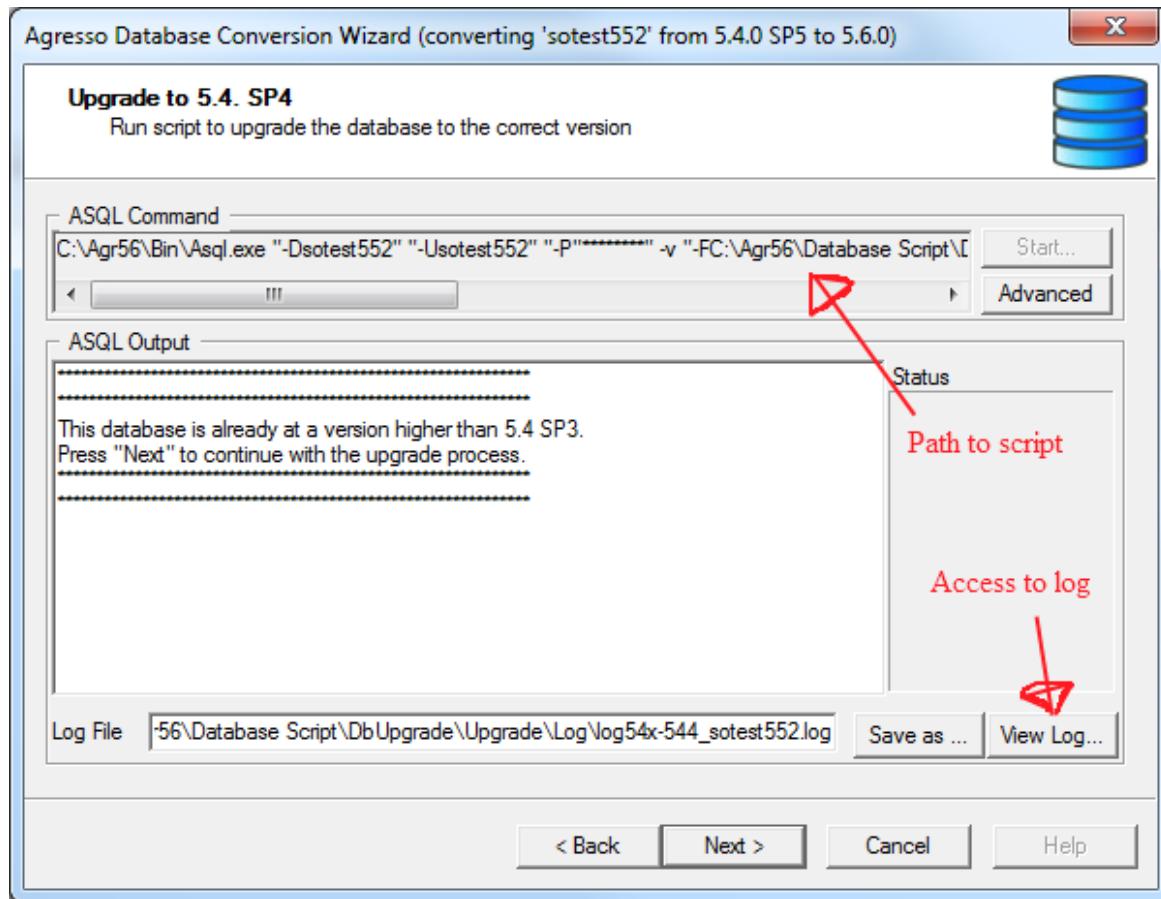
Location

The upgrade wizards are found in the 5.6 installation directory, at the following location:

`<...Agresso 5.6>\DatabaseScript\DbUpgrade\Upgrade`

Script details

The upgrade wizards run one or more scripts. In the wizard dialog, you will find the path to the current script. After the script has run, you can study the log:



The upgrade user

Temporary user for all clients

During the main upgrade (when upgrading from 5.4x only), a temporary Agresso user, **upgr55**, is created, enabling you to log on to Agresso, and set up roles and access rights for the existing users. This user is created for every client in the database, the password is upgr55.

When errors occur

It is important that all errors encountered during the upgrade process are reported back to Agresso.

Customer

As a customer you are asked to report the errors as tickets in the Heat system.

Partner or subsidiary

If you, as a partner or subsidiary, have access to the Heat system, report the errors as Heat tickets. Otherwise, please report the errors to the Technical Forum – Agresso 5.6 Technical Consultants <http://forum.agresso.no/>.

Do not report the problem if it is related to duplicates in the database, full disk, full tablespace or similar.

Test System Upgrade (all tasks)

UPGRADE OVERVIEW - TEST

Process description

The upgrade process presented here, starts when you have a valid 5.4 SP2(+) system in your test environment. For database upgrades, see Upgrade Process Overview.

The Test upgrade goes through four stages:

1. The Main system upgrade wizard performs all the required, basic upgrades of your system: It will add new tables, and convert old tables and data to the new formats - as far as it goes.
2. If relevant, you will run the Workflow wizard. This will convert data from Invoice Manager to the new Workflow solution, and also create new roles according to the the 5.6 Role structure.

Next, and this task is always required in stage 2, you must establish the new authorisation structure, giving the users access rights in accordance with the new Role concept. During upgrade og access rights and roles, you may also want to modify the roles automatically created by the workflow wizard.

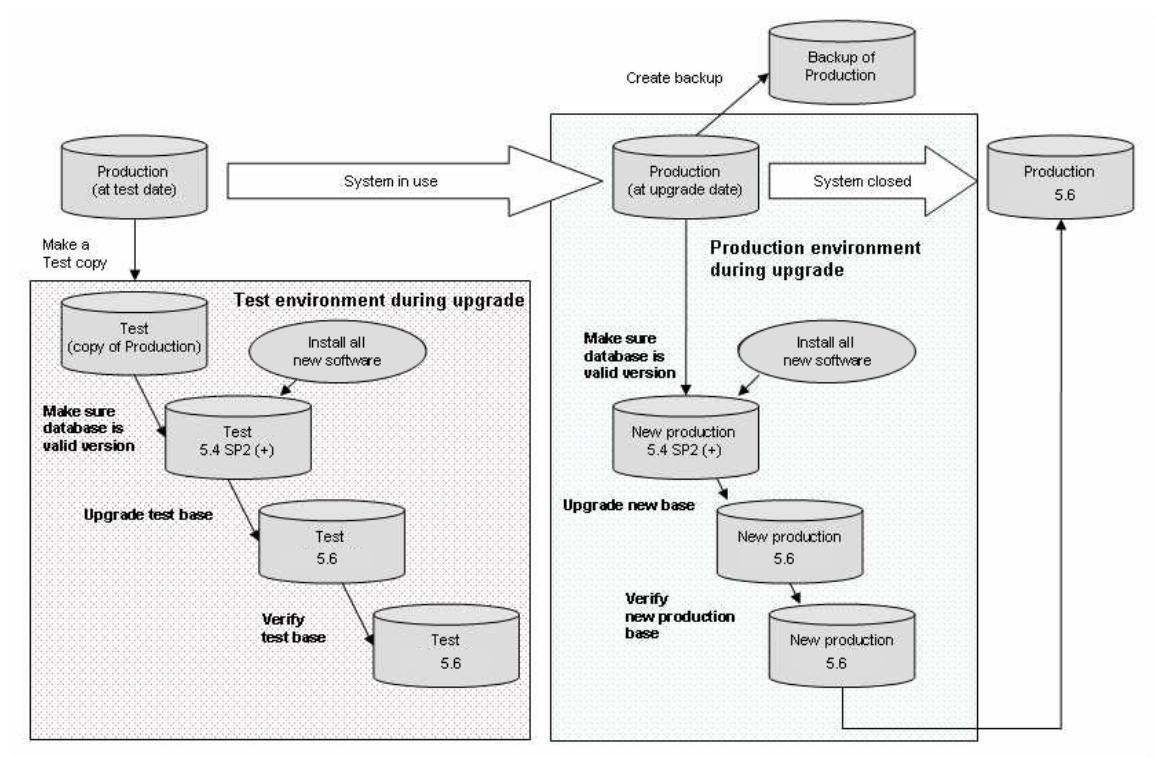
Note: You can either create the new role structure from scratch, or you can use a wizard to convert the old groups, web roles in one go. See [Authorization and Roles](#) for details.

Further - you will need to perform upgrade tasks for the remaining system areas used by your installation, such as Document archive, Data control and Invoice follow up (the latter is handled by Action overview in ABW 5.6).

3. You must run upgrade wizards and perform manual setup for a few functional areas (or modules), where new implementations in ABW 5.6 prevents automatic upgrading. Modules not in use can be ignored.
4. Finally, you finish the upgrade by cleaning up browser templates and remove tables not longer in use.

Overview diagram

The diagram below summarises the main tasks required for a complete upgrade - intended for the Test environment:



Manual setup tasks

There are several places where we have explicitly stated that manual setup is required - in addition to - or as an alternative to - running a wizard.

Some of these tasks may take time, and can in practice go on for several months, and long after the subsequent tasks, as outlined in the diagram above, has been completed.

Please note: The manual setup tasks is not described in these Technical Guidelines (you will see the details when you enter the detailed descriptions). The required documentation is found in the relevant Release Notes for the area in question.

Important note

If you plan to use the **Copy Setup Between Databases** wizard to transfer users, roles, menu access and optional setup from Test to Production, be aware of the following:

The wizard will copy all users, passwords, roles etcetera from the test (source) database to the production database. The production data will be replaced with data from the source database, with two important consequences:

- New users in the production system (after the test upgrade started), will not be transferred - unless you make sure that they also are registered in the test system.
- Passwords changed in the production system after the test upgrade started, will be overwritten by the old passwords.

Main System Upgrade

PREPARE THE UPGRADE

Install 5.6 software

When your Test database is in the correct 5.4 version, you should install all 5.6 software.

Verify disk space

Available disk space needed for the upgrade is approximately the size of the largest table + 10%

Note: Expanding the maximum number of characters allowed in a field does not have any immediate impact on the database size. The expanded columns were previously of data type char, while they now are redefined to varchar.

Create 5.6 Data Source and initialise Business Server environment

Use the **Agresso Management Console** to create a new 5.6 Data Source connected to the old database.

When the connection is up and working, you must also initialise the Business Server environment (select the **Business Server** node in AMC and then **Initialise Business Server**).

Rename or delete amendment tables

Logging

If amendment logging is turned on for an Agresso table, an amendment table (or shadow table) are created and continuously updated with all table changes.

Note: During upgrade, all amendment logging will be turned off (by the wizard), and must be manually reset by using the ABW Smart client.

Table for amendment tables

All amendment tables are defined in the table [aagamendlog](#).

Naming standard: An Agresso table name is constructed from the structure
`a<module><identification>` (example: `a_cr_client = acrclient`) while an amendment table is extended with the letters `shd` between `<module>` and `<identification>`.

If you turn on amendment logging for `acrclient`, the amendment table `acrshdclient` will be created and added to [aagamendlog](#).

Old amendment tables

During previous releases, the amendment tables have not been upgraded. When upgrading to 5.6, these tables will be checked, which may lead to a large number of errors.

Recommendation

To avoid problems with existing amendment tables, we recommend that you rename (or delete) all amendment tables before you start the upgrade process. Thereby, the upgrade wizard will not find them, and they will consequently not generate any errors.

If you need the amendment tables for historical reasons, they will always be available in your database copy.

To remember

Custom object definitions

The main system upgrade wizard will initially check the database for any custom objects, and write a detailed report containing all the old definitions (scripts). During the upgrade process, all custom objects will be removed.

The generated report, named [*user_def_obj.txt*](#), will be saved to the default log directory. You will need this report, as well as a detailed knowledge of the new table definitions (see Table changes in version 5.6) to restore previous functionality in Agresso 5.6.

The *custom objects* that will be removed, are:

- triggers and indexes,
- procedures,
- shadow tables,
- user defined views.

RUN THE UPGRADE WIZARD

The upgrade wizard

You find the upgrade wizard at the following location:

[...Agresso 5.6\DatabaseScript\DbUpgrade\UpgradeWizards.exe](#)

Upgrade scripts and log files

Logging and error handling

The **Log File** will, when the script has run, contain a description of what happened. Each statement are listed, along with status after execution.

Note: You should - as a general rule - always look at the log file when the script is finished. If any errors occurred, they must be corrected, and the script must be run again - *before* you continue with the next upgrade step.

When we describe the actions for each step below, we *do not* ask you to view the log files. We take it for granted that you will do so!

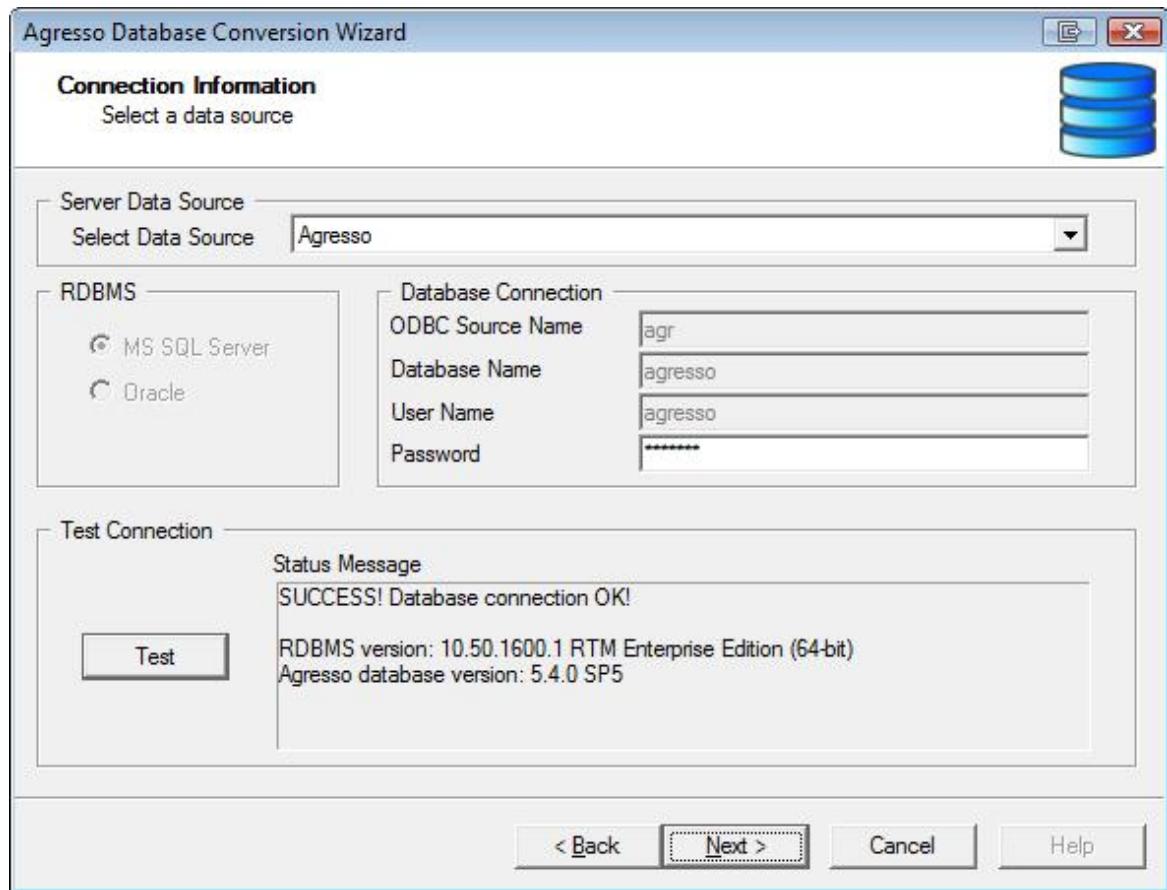
Logging details

You can define the level of details to be listed in the log file by using the **Advanced** button available in the various wizard dialogs.

Run the upgrade wizard

1. When the upgrade wizard is up and running, select **Convert from 5.4**. Then select Step 1 [*Main Upgrade Wizard*](#). The **Connection Information** dialog will be displayed.

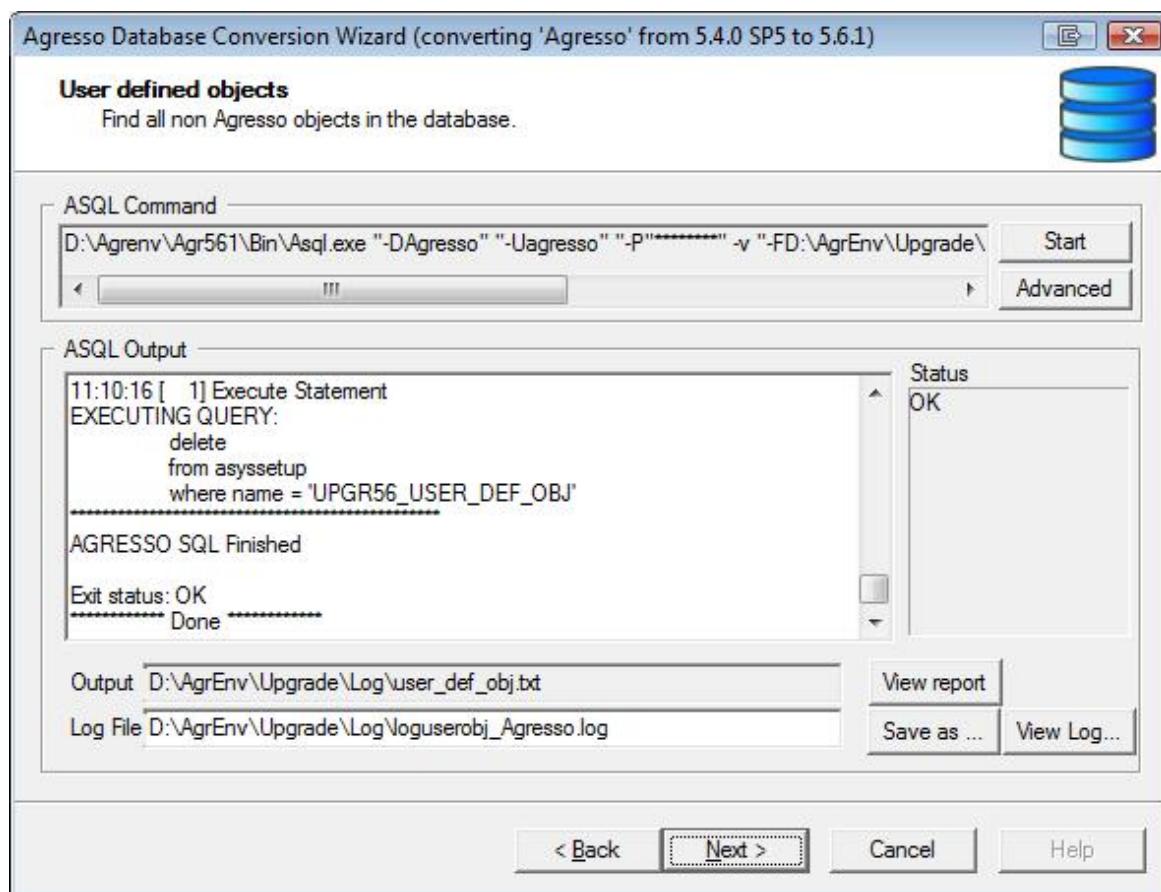
 [Upgrade Wizard - Connection Information](#)



2. Select the data source, enter password, and click **Next**.

Note: If you get an error message, just accept the proposed action - and make a note of it! The error is not significant for the upgrade, but it will be when running Agresso later.

[Upgrade Wizard - User defined objects](#)



More step 3 details

Note: The report containing the custom object definitions are shown in the **ASQL Output** field. You will need this later.

Important: Do not continue if any errors are found in the database. These may be:

- **Null values** - these must be fixed before you continue. Use copyms/copyora to copy the table with the NULL values out, and then into the database again.
- **Duplicates**. Do not continue with the upgrade if there are duplicates in any of the tables
 - aaguser,
 - acruserinfo or
 - aaguserfunc.

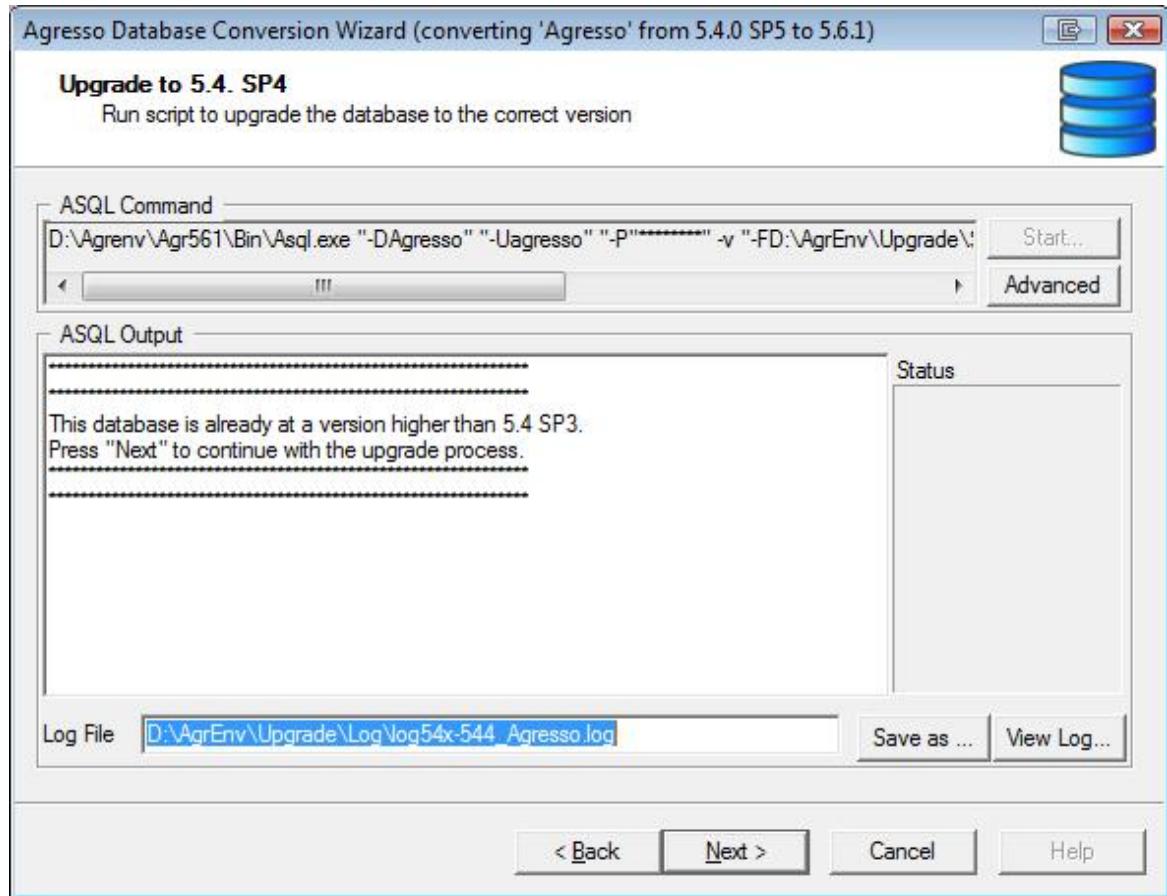
You must find the duplicates and remove them (or rename them).

Note: You must re-run the User defined objects step until there are no errors.

3. Click **Start** to run the script.

Important: You must correct all reported errors and make sure that everything is correct (re-run the script - until no errors are reported), before you can proceed to the next step.

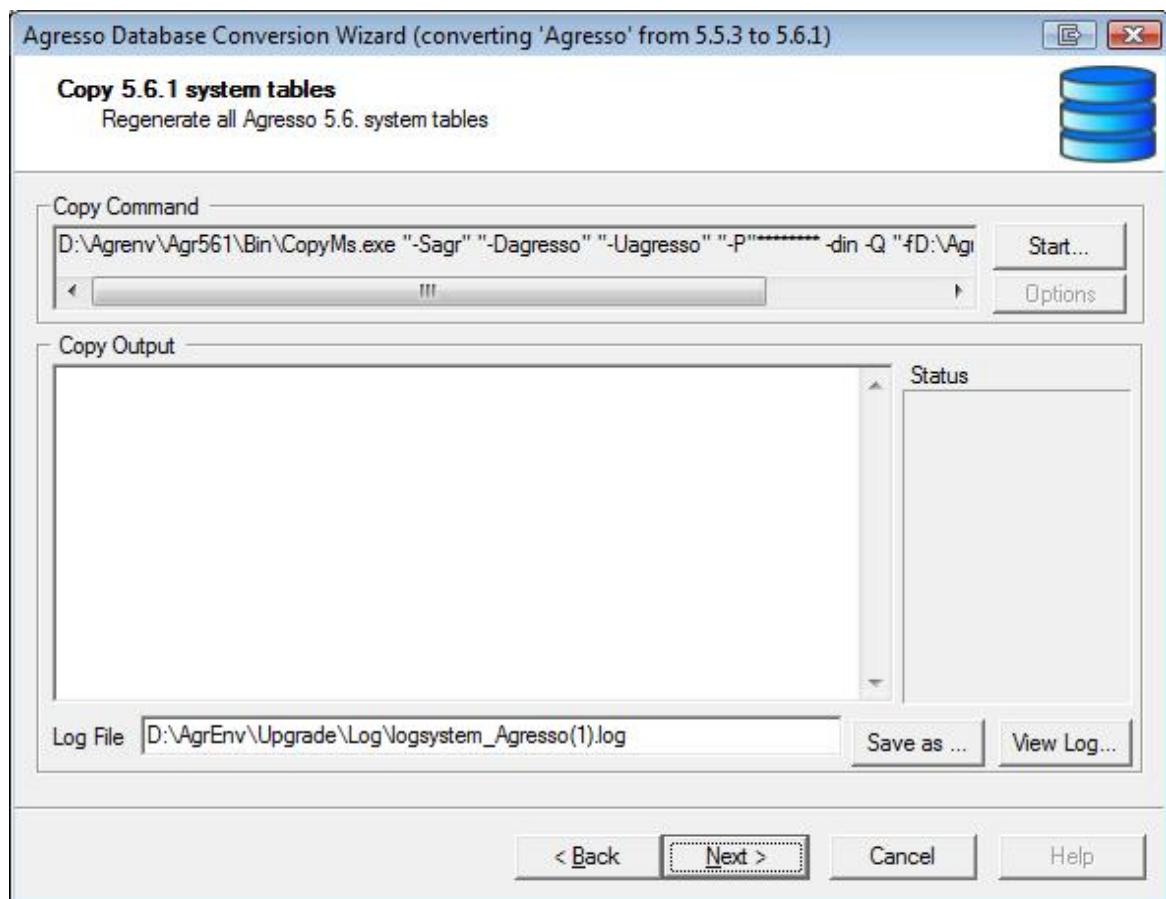
Upgrade Wizard - Upgrade to 5.4 SP4



4. If your database needs upgrading, click Start. Click Next to continue.

You are ready to restore the Agresso system tables, some awf* tables, and dictionary tables.

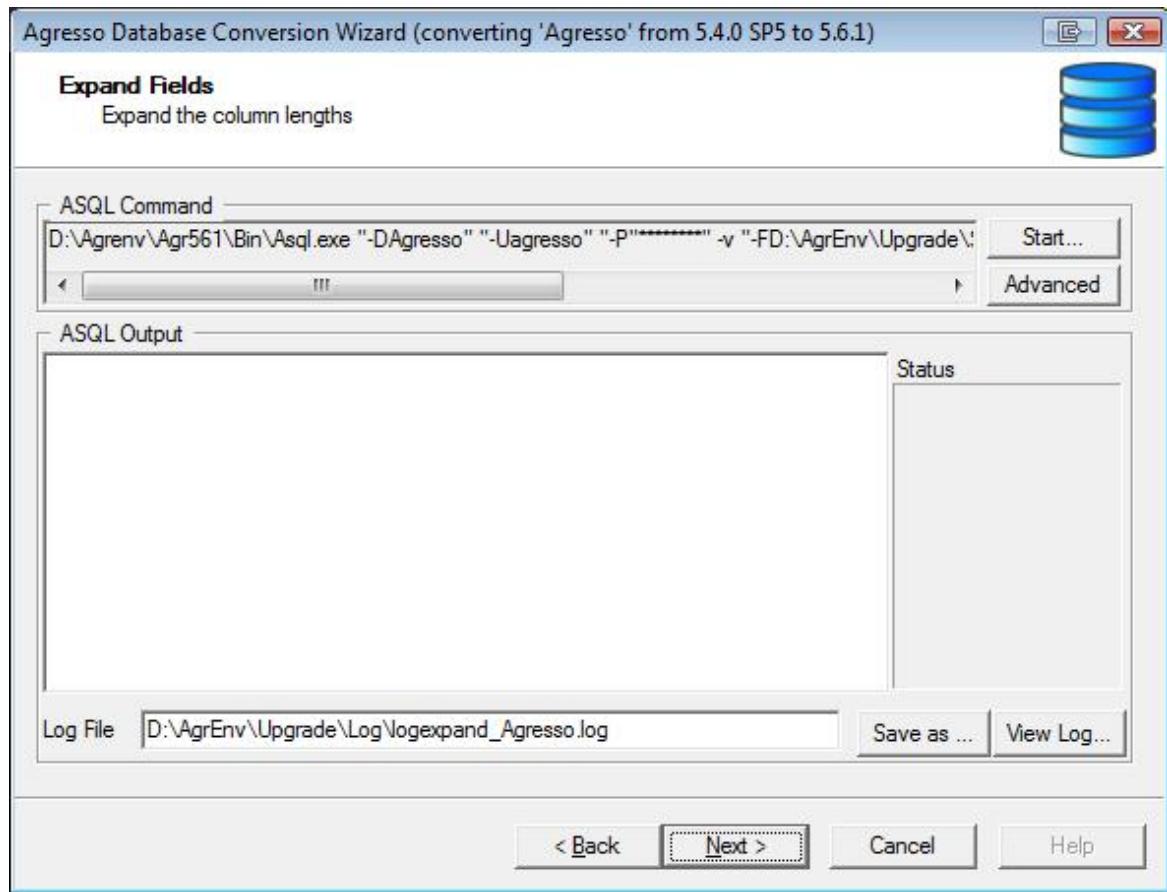
Upgrade Wizard - Copy in System Tables



5. Click **Start** to run the script, then **Next** to continue.

Next, the table columns will be expanded:

Upgrade Wizard - Expand Fields



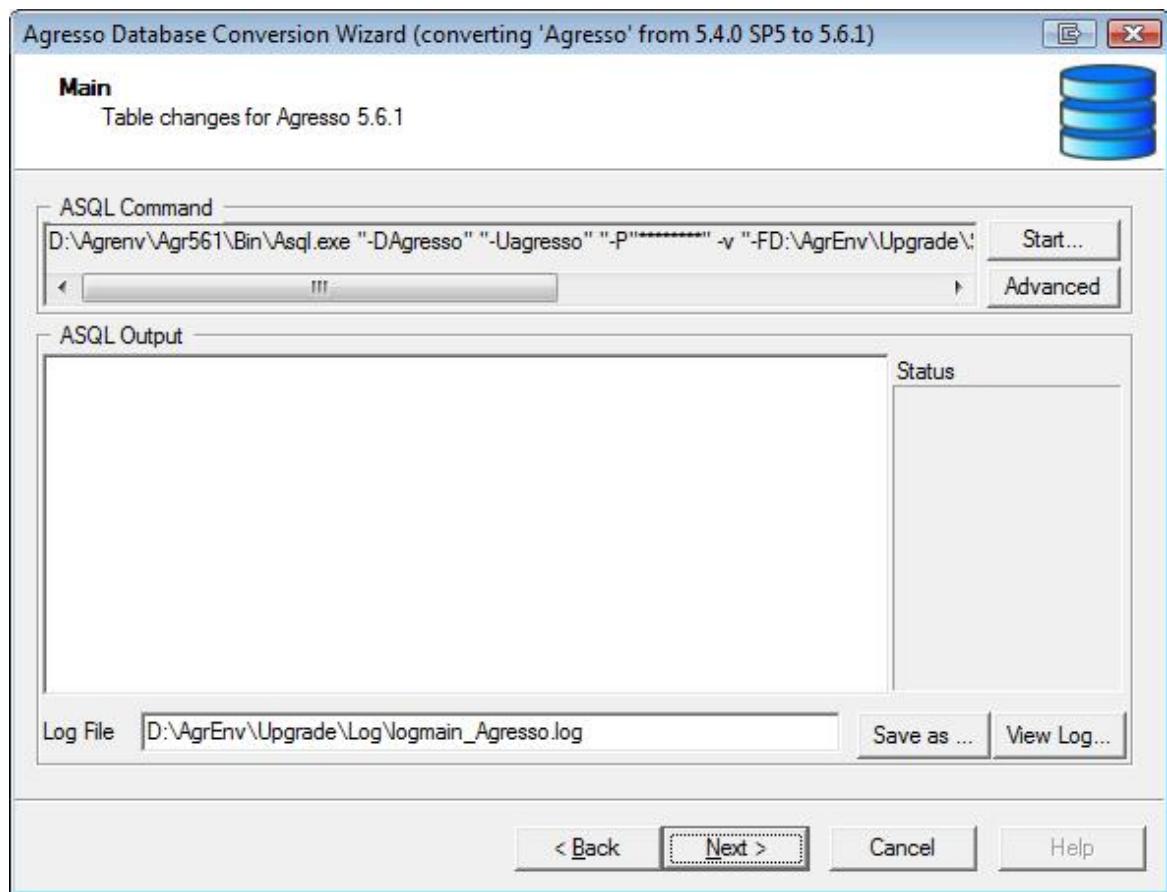
6. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

Note: This step might take some time.

Duplicates should be removed: If duplicates are found, it is recommended that the duplicates are removed before you continue. You do not have to rerun this step after the duplicates are removed.

You can now introduce the main changes in the database structure:

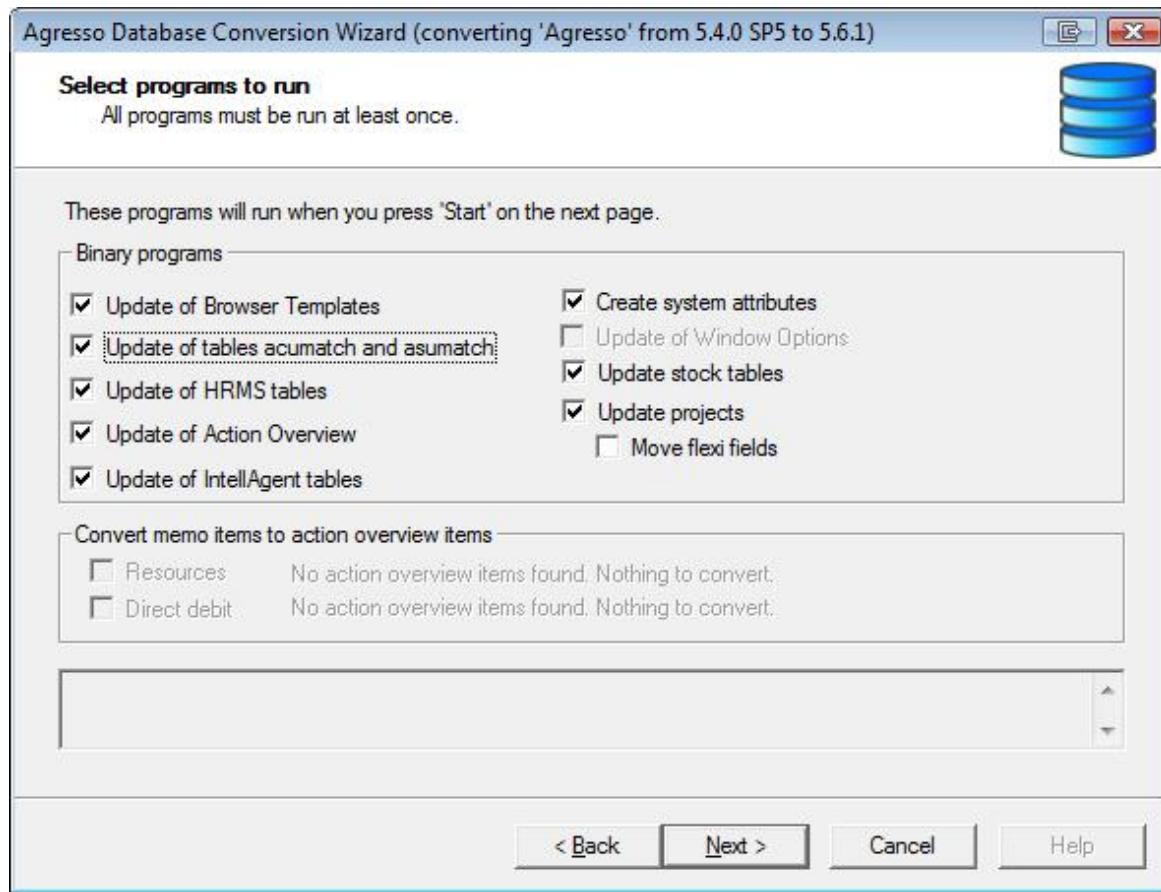
[Upgrade Wizard - Main - Table changes for Agresso 5.6](#)



7. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

The final changes to the database structure requires that you run some additional programs.

Upgrade Wizard- Run binary programs

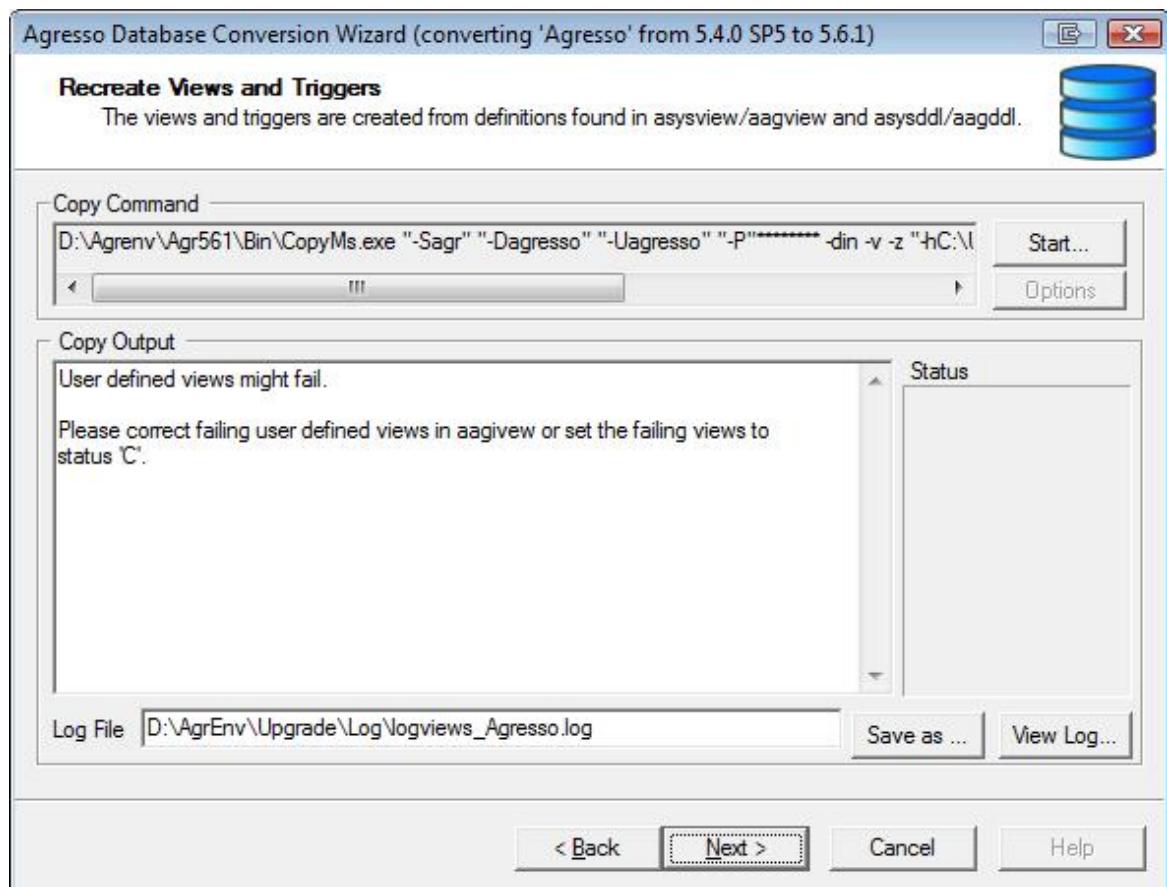


8. Select all programs listed and click **Next**.

Note: All programs must be run , but not necessarily at the same time. It will not do any harm if they are run more than once.

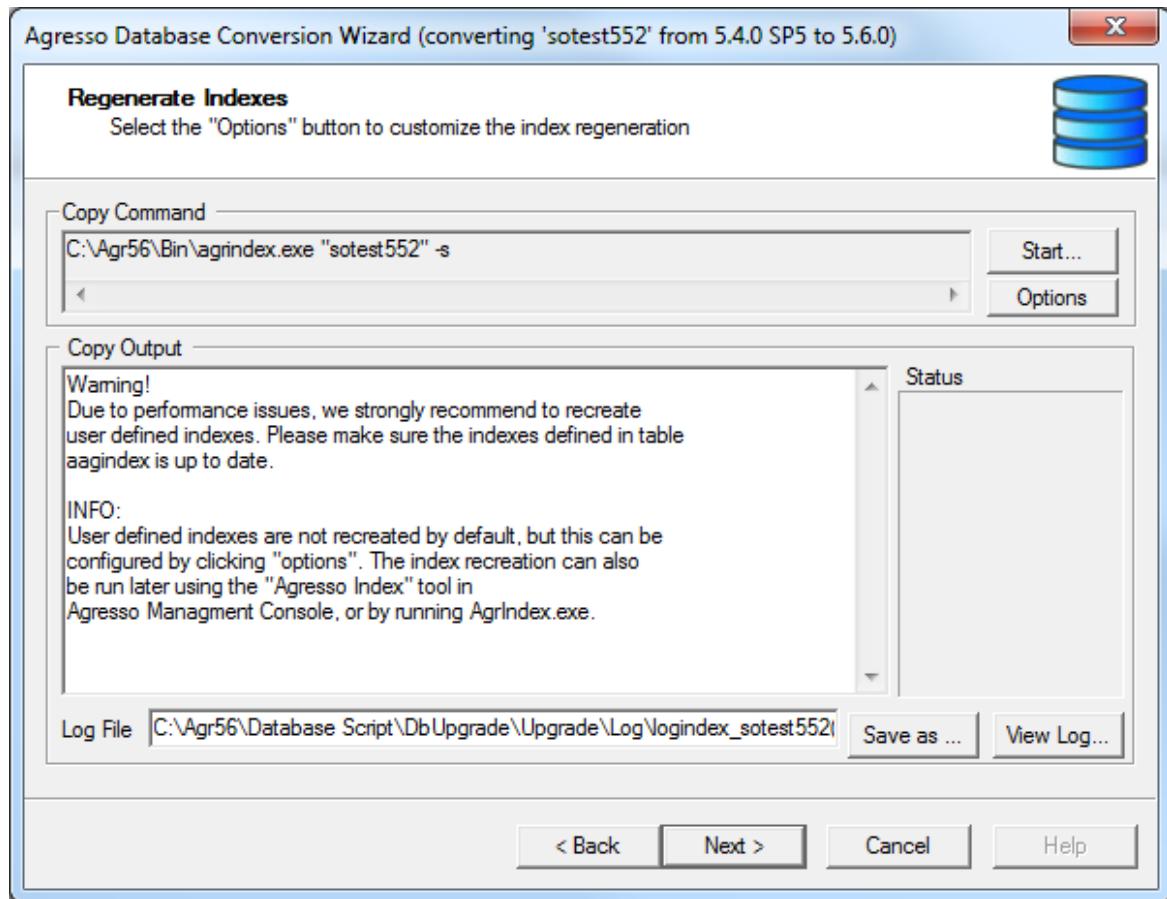
9. Click **Start** to run the selected upgrade programs, and then **Next** when the programs are completed.

[Upgrade Wizard - Recreate Views and Triggers](#)

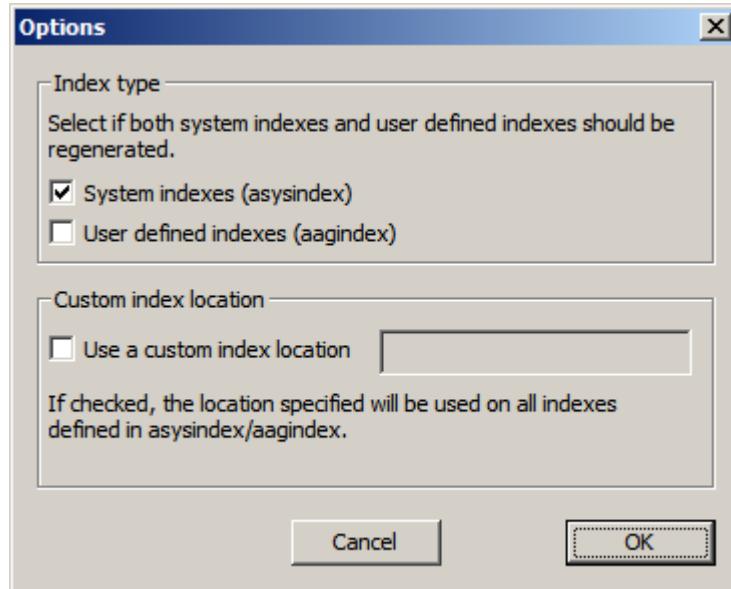


10. Click **Start** and **Next** in the next windows, in order to restore system views and indexes. This will bring you to the final step.

 Upgrade Wizard - Regenerate Indexes



11. Click the **Options** button to open the **Options** dialog:



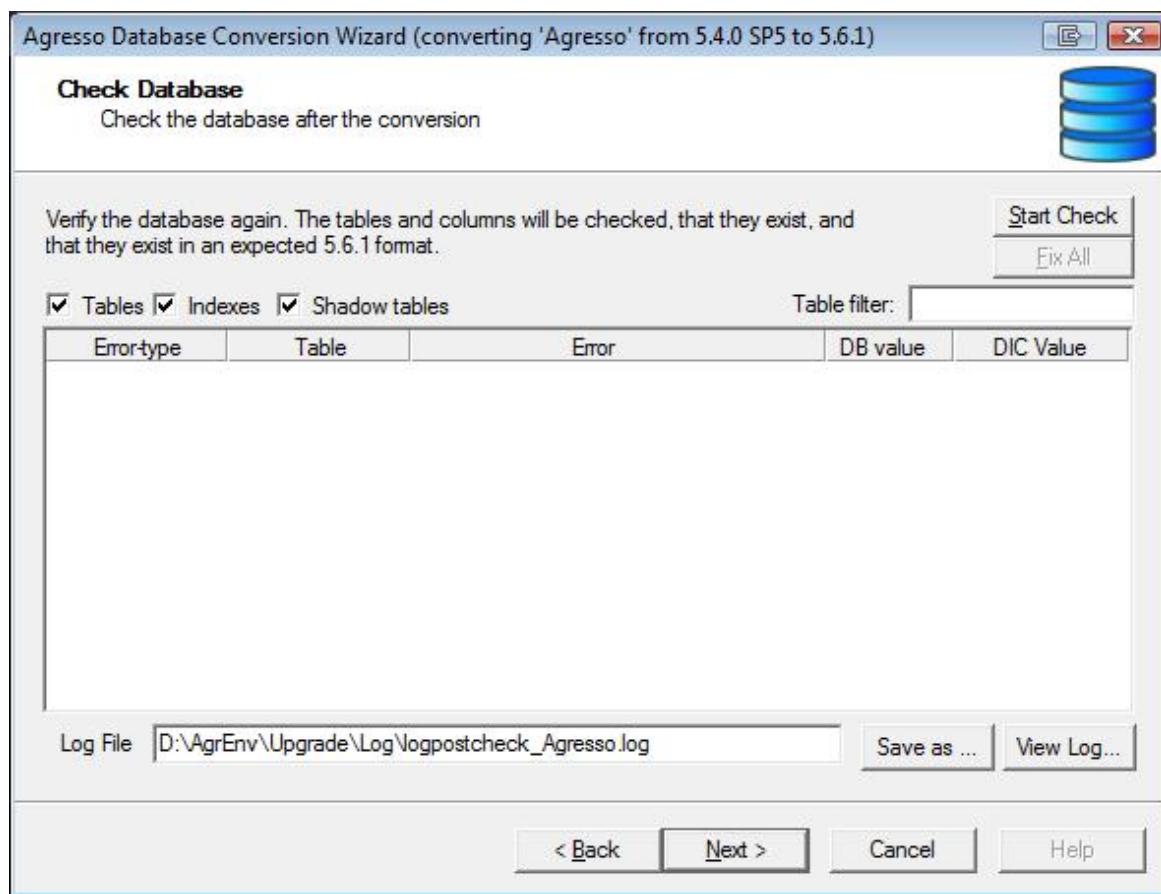
A note about indexes

By default, user defined indexes (stored in [aagindex](#)) are not re-created. Due to table changes, you should also change these, to avoid reduced performance. For details, see [AgrIndex](#).

12. Select options and click **OK**. Then click **Next** to continue with next step (Check Database).

This check might take some time.

Upgrade Wizard - Check Database



13. Click Start Check.

This check compares the table structure, tables, columns, index with the format defined in Agresso's agrsys* tables.

If any differences occur, double-click on the error line to correct it. Or try the Fix All button. The Fix All might take some if tables with a lot of data needs to be fixed.

It is important to have the database with all the tables, columns and indexes in an expected format.

If there are tables with duplicates, clean up the duplicates and regenerate the index. A missing index might be crucial for the performance.

The following differences can be ignored:

Columns in the table are not found in the dictionary.

Columns are longer than expected.

COMPLETE THE MAIN UPGRADE

Prerequisites

The output file [*user_def_obj.txt*](#) will contain all necessary information for restoring user defined views and objects. It will also list all duplicate user Ids (Oracle only).

In order to restore previous, custom functionality, you will require [*user_def_obj.txt*](#), but also a detailed description of the 5.6 database structure. See relevant Table changes in the Appendix (Agresso Data Dictionary).

Tasks

To complete the main system upgrade, you should do the following:

1. Restore user defined objects and views

2. Register license

Restore user defined objects and views

A note about amendment logging

During the upgrade, all amendment logging are stopped.

Update tasks

The table below describes how custom objects can be upgraded:

Object type	Update tasks
Triggers and indexes	Use a database tool to change the columns and data types to fit to the new database structure.
Procedures	Use a database tool to re-define the procedures according to the new database structure.
Amendment logging (shadow tables)	Use the Smart client to open the AG30 Activation of logging server from the System Administration Data Control Amendment logging , and re-enable logging.
User defined views	Use the Smart client to open AG17 Database view definition . Zoom into each query, make the necessary changes, and remember to save the changes to have the views recreated.

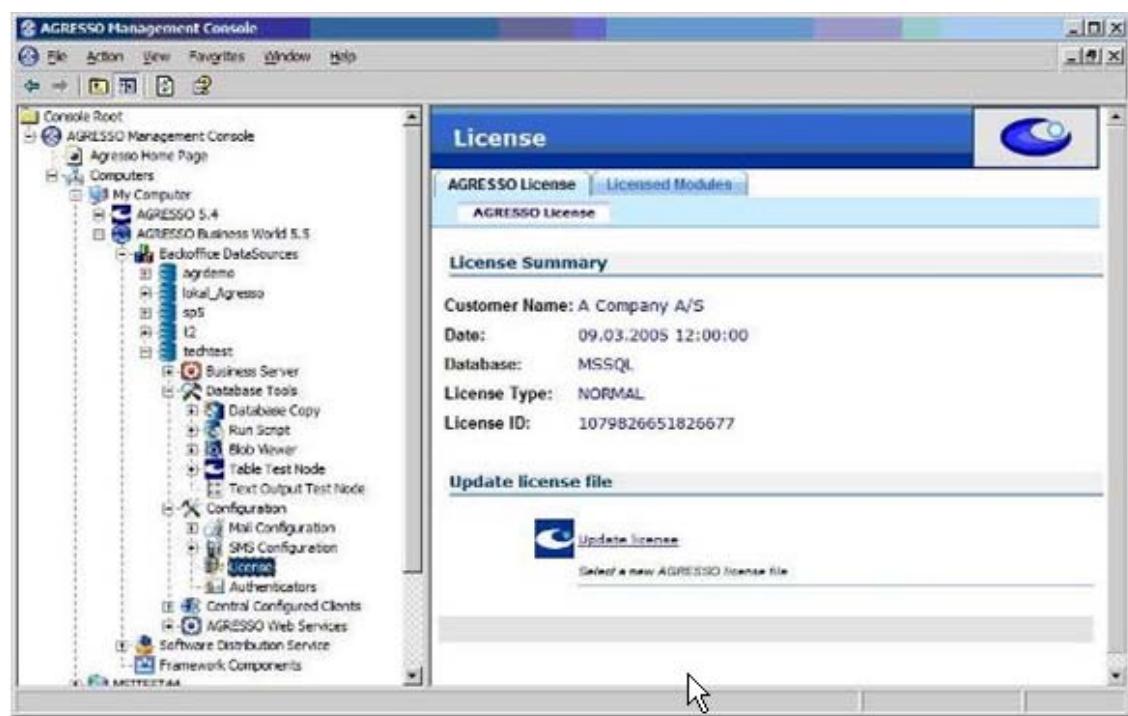
Register license

View licence

The license node in AMC allows you to add a new Agresso license to the database. It also enables you to view the licensed modules and the number of users registered for each module.

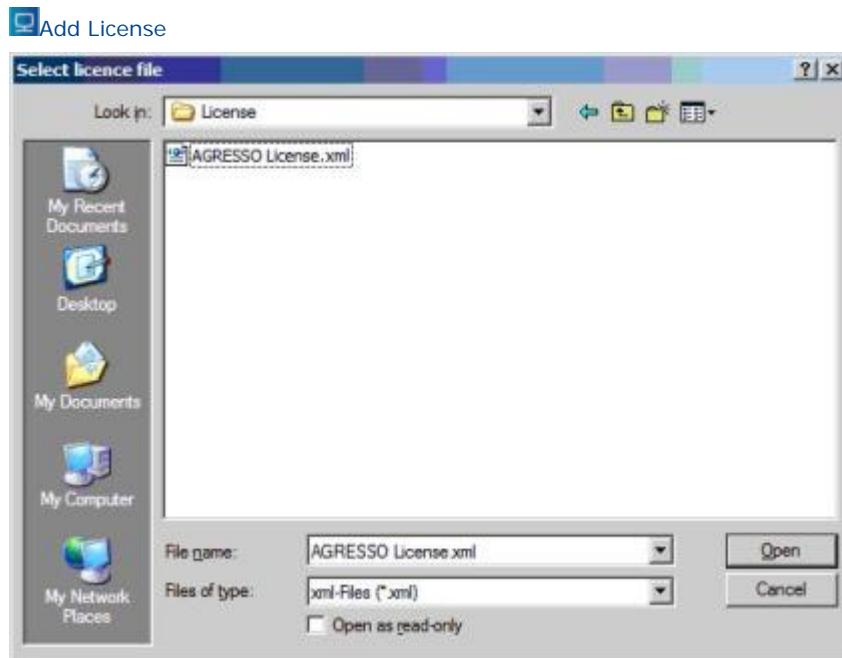
If you have a demo license installed, this node will inform you if the license will soon expire, or if it has already expired.

View License



Add License

To add a license you can either right-click on the license node and select **Add License** from the **All tasks** context menu. Or you can click on the [Add License](#) or [Update License](#) link found in the right pane of the management console. Then you will be prompted to select a license file.



You can also add a license using the Agresso Client Configuration tool.

Note: You need to enter a *new license* when upgrading from 5.4. to 5.6 to make all parts of Agresso work. If required, contact your local Agresso support office.

Login info

Your are now ready to test your Agresso Business World application with the following credentials (Note: **upgr55** is correct name when upgrading to 5.6):

Username:	upgr55
Client:	<your client>
Password:	upgr55

Additional System Upgrades

You can now proceed with Additional system upgrades and setup.

Additional System Upgrades and Setup

INVOICE MANAGER AND WORKFLOW

Agresso Workflow

New workflow solution

The workflow module (introduced in version 5.5 of ABW) replaces the Compello system. The new workflow module is based on an entirely new data model. Some Compello tables are kept to show historic workflow maps and are renamed to `ac<compello table name>`.

Wizard parameters

The Workflow upgrade wizard converts the data in the Compello tables to the new data model and defines processes in the new workflow system to match the existing set up in Compello.

Note: The wizard also makes it possible to upgrade rules from Compello. This is, however, NOT recommended, as the new workflow solution allows you to create far more advanced and generic rules than before.

Parameters

During the upgrade, you will be prompted to fill in some necessary parameters (see 3. below). These are explained as follows:

Parameter	Default value	Description
Client	*	The Agresso client to upgrade. Wildcards can be used (for example: *)..
Run as user		The user that will run the upgrade wizard.
Date From	19000101	Format: YYYYMMDD. Older document, transactions and images will be ignored.
Date To	20990101	Format: YYYYMMDD. Newer documents, transactions and images will be ignored.
Convert Invoice Manager registers	Off	Select if fixed registers from Compello shall be upgraded. If checked: <ul style="list-style-type: none"> • Users – Users will be created and the workflow flag will be checked for them. • Groups – will be converted to Roles • Substitutes and supervisor definitions
Convert Invoice Manager rules	Off	Only selectable if the Convert Invoice Manager registers is selected or the registers were upgraded in a previous run of the wizard. If selected, all Compello rules will be upgraded to the new rules tables. Note: This is not recommended (see The upgrade wizard above).
Invoice Manager maps	Off	If selected, historic transactions and Compello map data will still be available. The old maps can be viewed from the new map viewer.
Create template workflow processes	Off	If selected, all template workflow processes will be copied to the clients chosen for upgrade. The template processes are drafts and will need to be committed by the user chosen in Run as user (see above) to become active. Note: This will overwrite any current workflow processes for the clients being upgraded.
Convert Invoice Manager archive	Off	If selected, Compello images will be upgraded to the new solution, and UNC-path (below) will be enabled.
UNC-path		UNC path to locate existing Compello documents. NB! Must be UNC path! Example: \\compelloarchive\\images

Upgrade Invoice manager

Upgrade consequences

The result of the upgrade processes described below is that

1. the Compello COM application is completely removed,
2. users will have direct access to the file share.

Dangers: The Compello document location must be configured as a file share (UNC name) with full access for everybody using the document archive. The share can be hidden, but there is no way of stopping an advanced user from mapping the drive and accessing the documents.

In practice, no one needs more than read-only access to Compello documents, and the documents are scrambled to prevent casual browsing.

Protection: We have tried combine these requirements, and reduce the security risks, by creating a dummy Windows user, which Agresso will use to access the documents. The system parameters DS_NATIVEFILE_USER and DS_NATIVEFILE_PWD allows you to configure this user. The dummy user (and no other) needs full access to the file share. When you run the document archive system within Agresso, the rights of this dummy user are picked up. Any attempt to map the file share outside of Agresso will be denied.

New document archive and old images

With the new document archive, customers with historical Compello images can select one of the following options:

- Leave existing documents in the Compello folders as read-only, for historical access only
- Move the documents into the Agresso Document archive (and remove them from the Compello database).

Required upgrade

To be able to view Compello documents in Agresso 5.6, you must at least:

1. Run the upgrade wizard to convert the database tables. (This can be done in stages by date etc.)
2. Configure permissions for the Compello file share.
3. Create a document type to access the Compello documents, for example Compello Invoices. This should belong to the GL Transaction key, and have the document system Compello selected.

This will change the pointers to the data only. The image files are not moved.

Complete migration

You may want to make a full migration for the following reasons:

- To have one document type only for all invoices, both old and new
- To be able to use the new Agresso document archive features

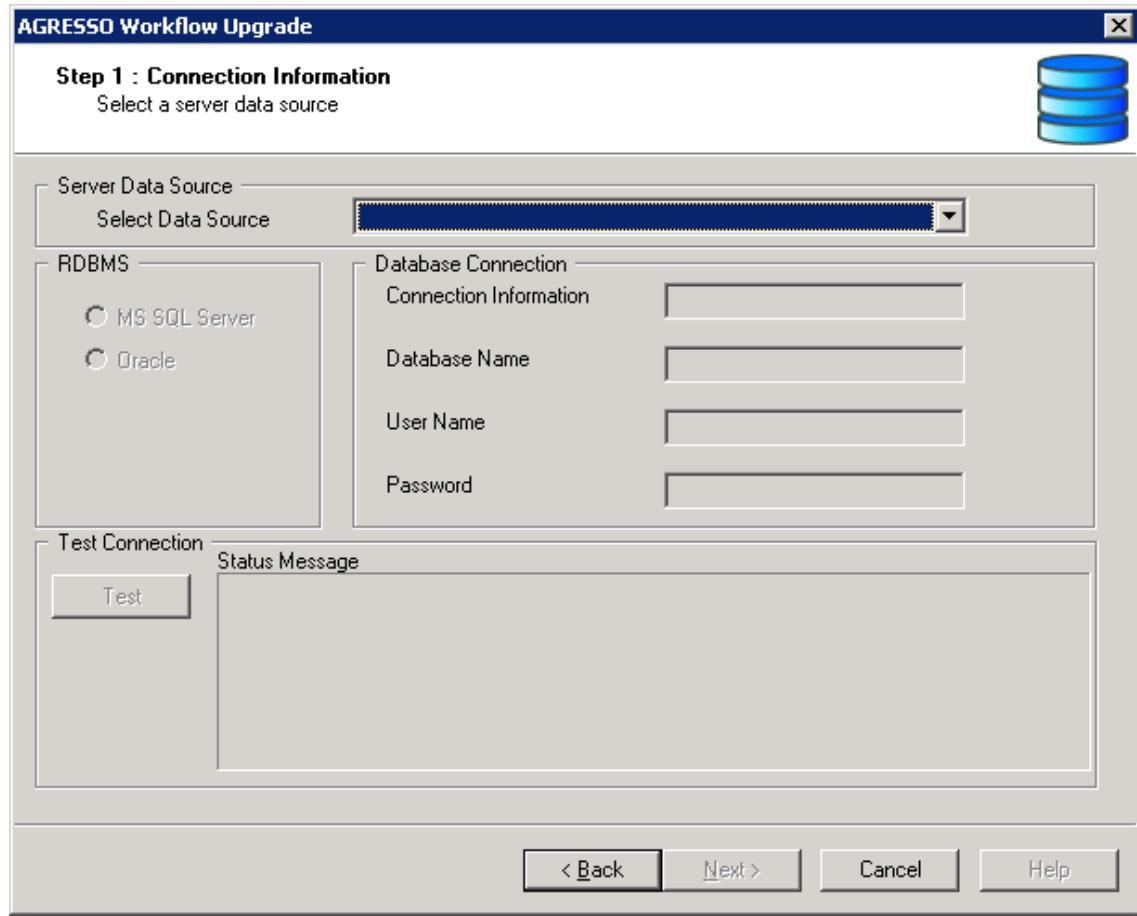
If you prefer a complete migration and remove the Compello archive, you can use the server process **DS01 Document copy** to move documents between document types. If the document types are based on different storage systems, as in this case, it automatically uses the drivers to do the conversion.

Run the wizard

Do as follows:

1. Start the upgrade wizard and move to the Step 1 dialog:

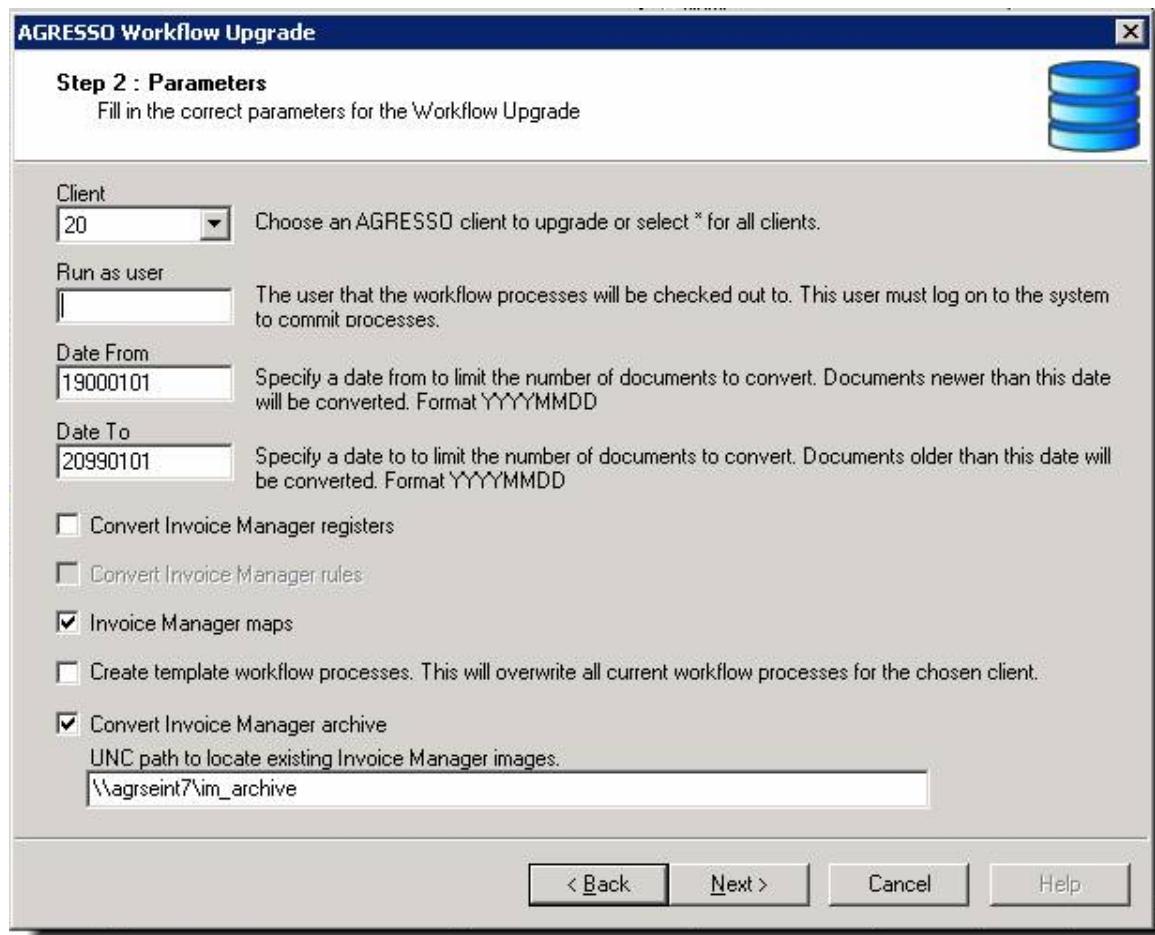




2. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

You are now prompted to set necessary parameter values.

Parameters - Step 2



3. Make sure the parameter settings are correct, and click **Next**.

You are ready to run the upgrade scripts.

4. Click **Start** to run the script and the **Next** to continue.

The (basic) upgrade is completed and you can close the wizard.

AUTHORIZATION AND ROLES

The ABW authorisation structure

ABW access for users and roles

A new and flexible structure for users and roles, with regard to access rights, was introduced in ABW 5.5.

An ABW user can belong to several roles, and the user's access rights will be the sum of all the roles' access rights. In addition, a user can have additional access rights independent of roles.

User type per client

ABW also supports the concept of user types. A user type is a definition of the user's general relationship to a specific client – independent of roles. This relationship is established by linking the user to another data register in Agresso (Resources, Customers), thus making it possible for Agresso to understand when a certain user is identical with a certain entity in the linked register.

Example: If user **myname** is set up with user type **Resource** and linked to the resource **12345** in **Resource master file**, Agresso will allow **myname** to enter hours worked for resource **12345**.

It is not required to define any user type, and for each possible user type *value*, only one specific relationship can be defined per client. The same user can have different user type relationships to different clients.

Resources only: Currently only **Resource** is fully supported as user type.

User type values

A user type is defined by linking the user to a registered entity in one of the Agresso master files. The table below describes the valid user types and linked registers:

DOCUMENT ARCHIVE UPGRADE

New Document archive solution

The Agresso Document archive was completely rewritten with ABW 5.5. The document archive is now an integrated part of a series of Agresso windows, and documents of any type (.doc, .pdf, .jpg etc.) can now be attached to most of the Agresso object's.

These profound changes require that all the previous documents must be converted.

Please refer to *Agresso 5.5 Release Notes, Document Archive* for details about technical and functional aspects of the new solution.

Invoice Manager Reference

See the topic [Workflow and Invoice Manager](#) for information on how to interface your old Compello images with the new Agresso document archive.

Running the upgrade wizard

Necessary preparations

During the upgrade, you will need to make a few selections in order to proceed (see 3. below). The parameters you must enter are explained in the following table:

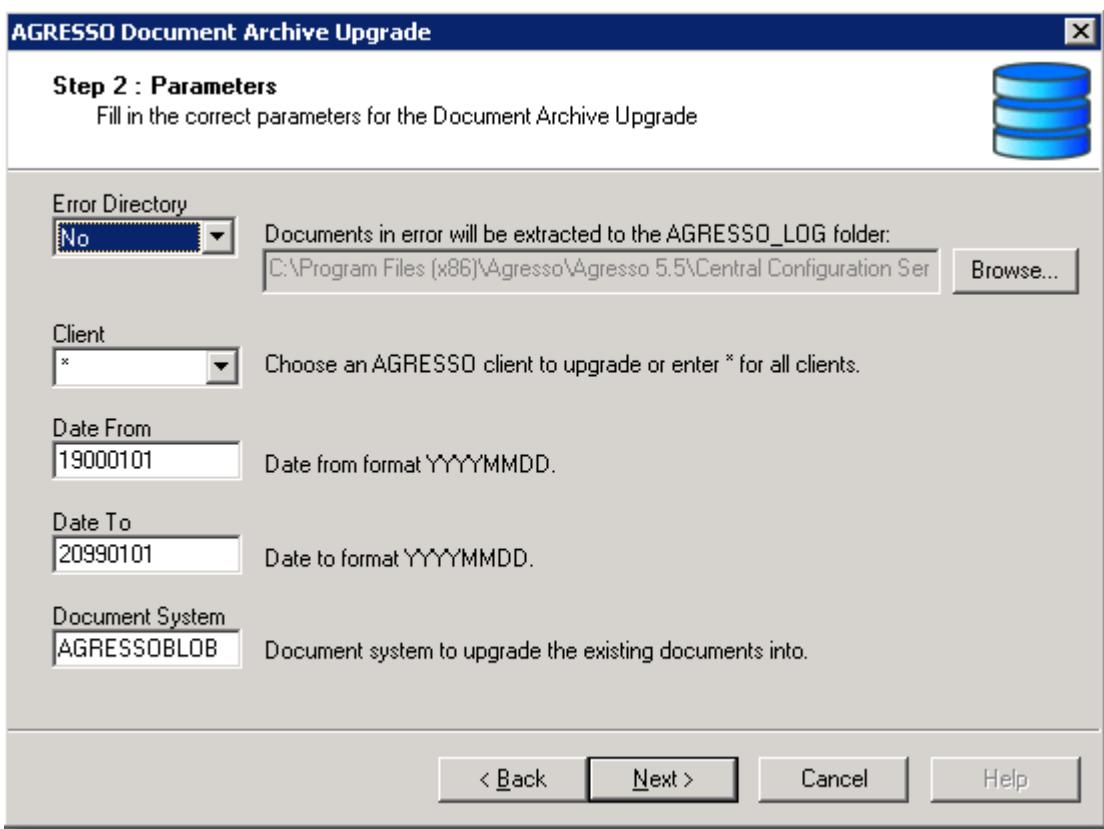
Field	Default value	Description
Error Directory	No	<p>Yes or No.</p> <p>If Yes, erroneous documents will be extracted to the server logging (AGRESSO_LOG) directory for manual fixing.</p> <p>If No erroneous documents will just be reported in the log (not extracted).</p>
Client	*	The Agresso client to upgrade. An asterisk (*) means all clients.
Date from	19000101	All documents registered at or after Date from will be included. Older documents will be ignored. Format: YYYYMMDD.
Date to	20990101	All documents registered before or at Date to will be included. Newer documents will be ignored. Format: YYYYMMDD.
Document system	AgressoBLOB	The new document archive can be connected to many document systems, including third party archives. You must choose a document system to upgrade the existing documents into. The default value is the new Agresso database archive.

Upgrade procedure

To upgrade to the new document archive solution, you will first start the Upgrade wizard. Continue as follows:

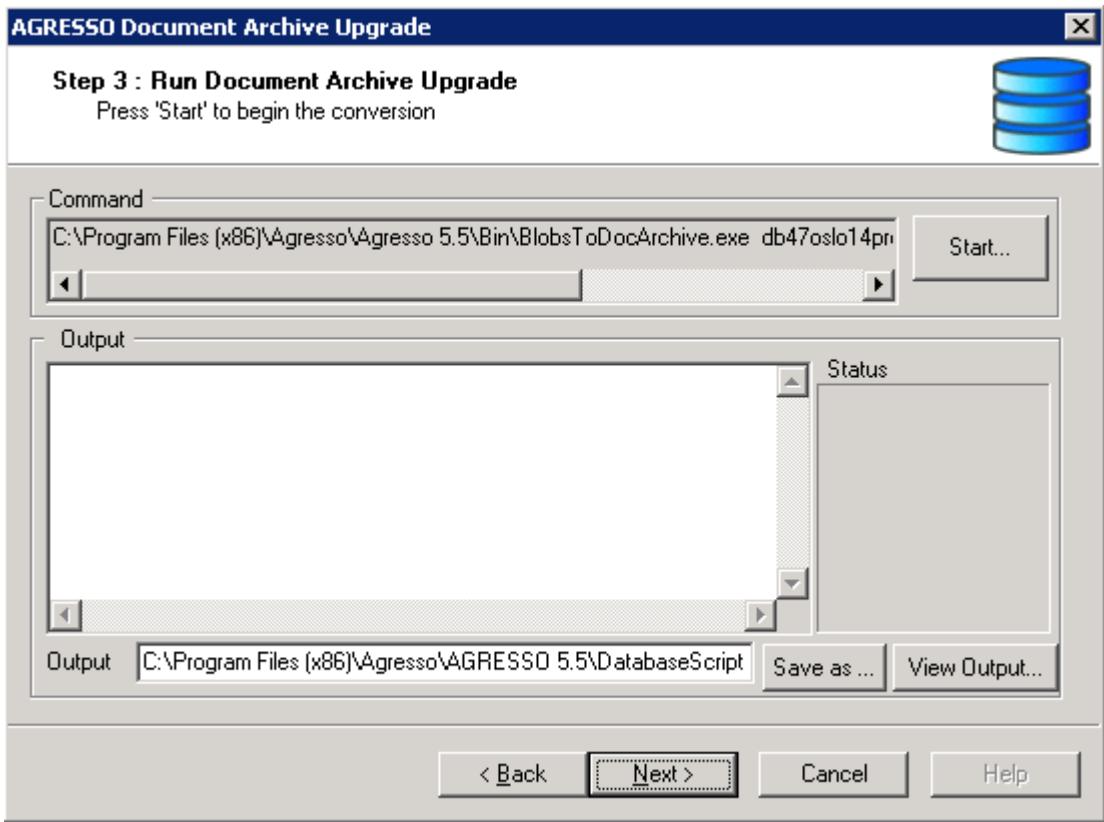
1. Double-click the [Document Archive](#) link in the Database Upgrade Wizards screen. After reading the introduction, click **Next**.
2. Enter connection information and click **Next** to prepare for step 2.





3. Fill in the required parameters and click **Next**.
You will now have to verify the path to the log file (output) made by the upgrade scripts.

Document Archive Upgrade - Step 3 #1



4. Make sure that the output path is correct, and run the upgrade by clicking the **Start** button.

5. When the conversion is done, do as follows:
 - a. Click **View Output** to study the conversion results
(we presume everything is in order)
 - b. Click **Next** to bring up the final window.

6. Click **Finish**

Note: If necessary, check the log file once more and make sure everything is correct.

MANUAL SETUP TASKS - TEST

The following areas require manual setup:

- From View options to Window options - See Release Notes for User Interface, ABW 5.5 SP1.
- Data control - see Release Notes for Common and System Administration, ABW 5.5 and 5.5.1 combined.
- Management of open items and Action overview - see Release Notes for Action Overview, ABW 5.5 and 5.5.1 combined

Functional Area Upgrades

RESPONSIBLES UPGRADE

Changes in data structure

The new solution for users and roles has required a completely new implementation of the *Responsible* concept in the Logistics module. Previously, the *algresponsible* table hold information about responsible codes and the *resource_id* (not the *user_id*) that was linked to the code.

Agresso 5.5 introduced a solution where a responsible belongs to a certain role, and where role membership is based on the *user_id*.

Upgrade of orders created by Agresso 5.4x

The main purpose of this Responsibles Upgrade, is to convert active orders (created in the various logistics modules) in such a way that the responsible person still can be identified when the order is further processed.

Note

The upgrade will *not* convert the old responsible resources into new responsible *roles*.

Upgrade tasks

The main upgrade tasks are performed by the responsible upgrade wizard, and are described as follows:

Task	Description
Automatic code matching	Match resource_ids from the <i>algresponsible</i> table with the 5.6 users. If a user can be uniquely identified by a resource_id, the upgrade wizard will establish a new connection between the 5.4 responsible code and the 5.5 user id.
Manual matching	For all unmatched resource_ids, you are prompted to manually enter user_ids. There is no requirement that the new user really takes part i 5.5 responsible role. This task can be repeated several times, until all (old) resource_ids have a valid match.
Automatic order upgrade	The wizard will search through all relevant orders and convert all orders with a responsible codes where there has been a sucessful match. Log: The result will be written to the log, and can be viewed in the table <i>algprophead</i> .
Manual order update	For all the orders that were not converted, you will have to use the registration windows for the various order types, to set the correct responsible. Note: This requires that the various responsible roles are set up in the

Responsible Setup window in the Smart client. See *Agresso 5.5 Release Notes, Logistics – Common logistics*.

Run the upgrade wizard

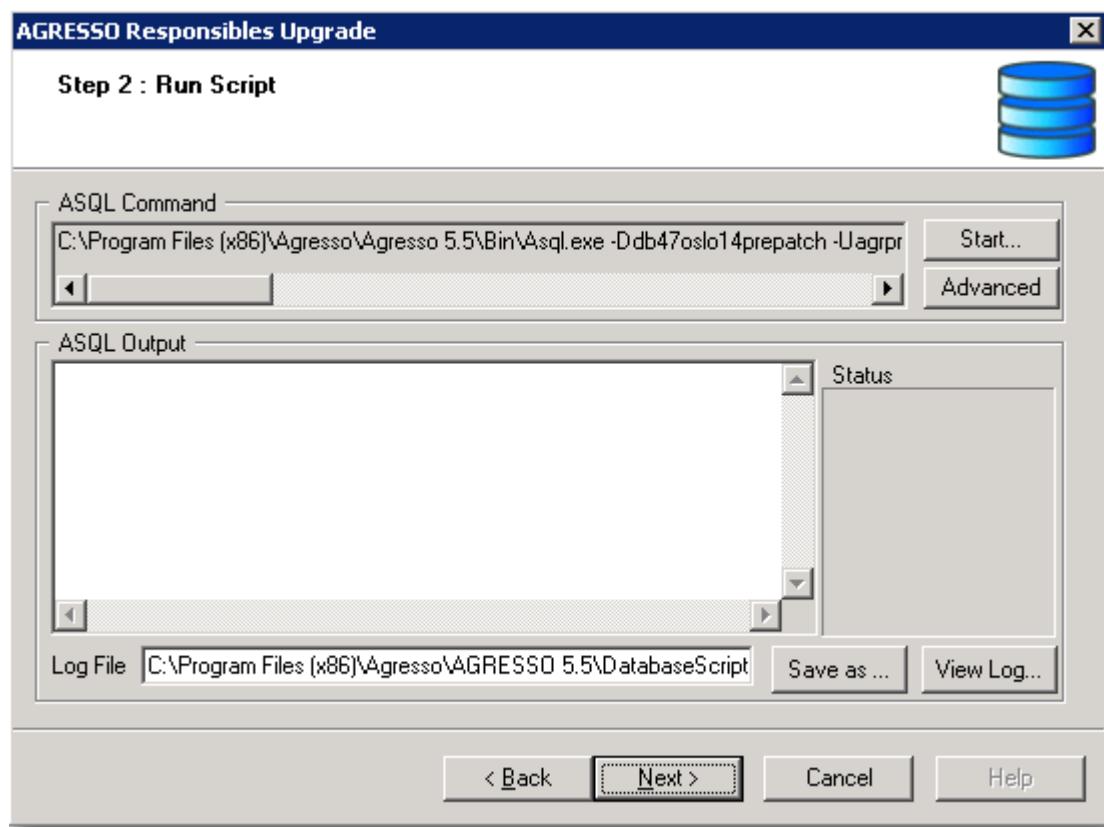
When the **Responsibles Upgrade wizard** is up and running, navigate to the Step 1 window, and do as follows:

1. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

You are now ready to run the upgrade scripts.

Note: If you already have run the upgrade scripts once, and in addition performed some manual matching (3. below), a re-run of the wizard will remove all manual updates!

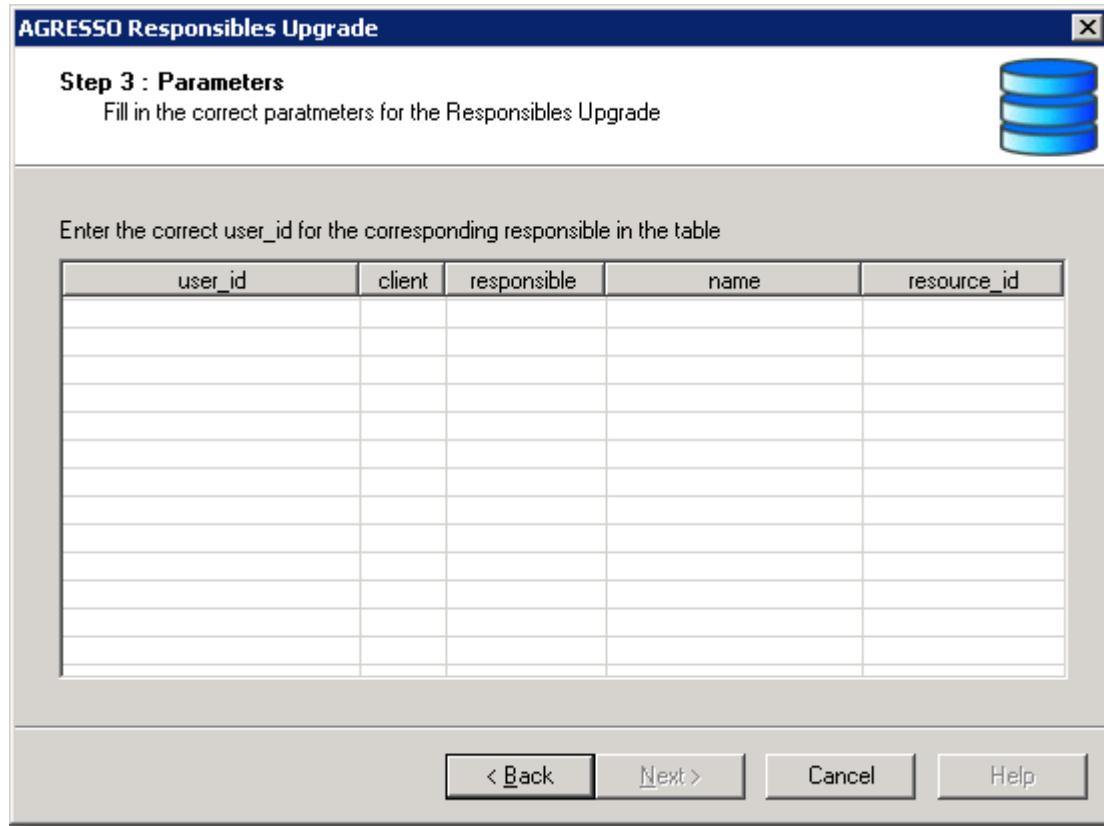
 **Responsibles Upgrade - Step 2**



2. Click **Start** to run the script, and then **Next** (we assume that everything went OK after **Start**). The wizard will match all identifiable resource_ids with the 5.5 user_id. When completed, the status for the *algresponsible* table is set to S.

You will now be prompted to manually enter missing user_ids:

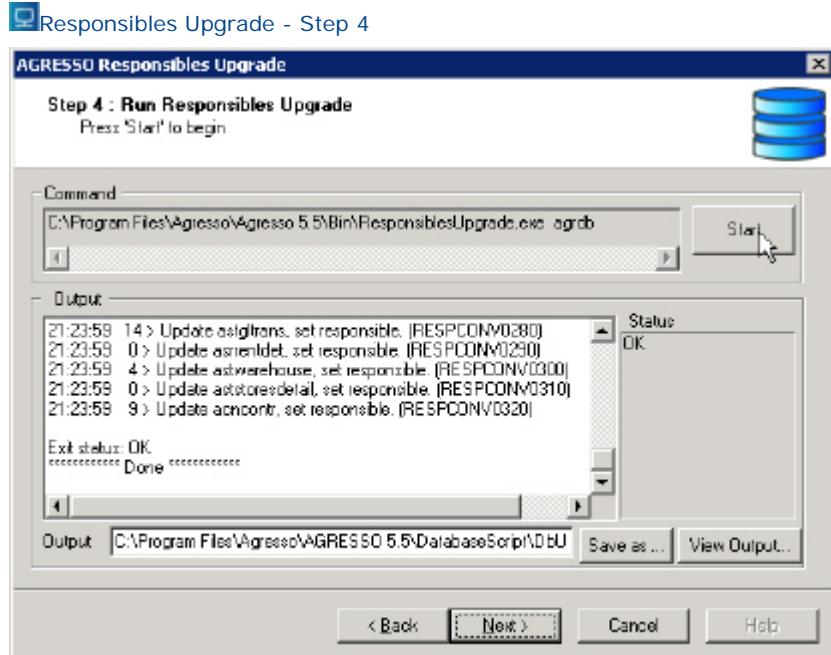
 **Responsibles Upgrade - Step 3**



3. Repeat steps a. and b. as many times as necessary:

- a. Add missing user_ids.
- b. Click **Apply**
- c. Click **Next** when you are finished.

The next task is automatic order upgrade.



4. Click **Start**, and then **Next** to finish.

This wizard can be run several times if you would like to add user_id's at a later stage.

Complete upgrade - additional manual tasks

There will normally remain some unmatched responsible codes on the orders. If so, the responsible roles must be registered in the Agresso Smart Client, and the orders must be updated in their respective registration windows.

HRMS UPGRADES AND SETUP

Wizards

You must run two wizards:

- [Balance Upgrade](#) and
- [Work Schedule and Absence Details Upgrade](#)

Manual setup

The manual setup tasks are described in the following documents:

- *Release notes, AGRESSO 5.5, HRMS/PCB integration*
- *Service Pack notes, AGRESSO Business World 5.5 SP2, Project*

PROJECTS UPGRADE AND SETUP

Wizards

To upgrade the Projects module, you will need to run the following wizards:

- [Balance Upgrade](#) - if not already run,
- [Work Schedule and Absence Details Upgrade](#) - if not already run, and
- [Timesheet Upgrade](#).

Manual setup

See the documents:

- *ABW 5.5 Reference manual, PCB Setup resources, time costs & income*
- *Service Pack notes, AGRESSO Business World 5.5 SP2, Project*

Finalise

FINALISE UPGRADE

Overview

To complete the upgrade, you should:

- Check the converted Browser templates, and correct any errors.
- Clean up duplicates and add indexes.
- Correct all user defined views due to the table changes (see Appendix, Agresso Data Dictionary)
- Remove all tables no longer in use.

Check and correct Browser templates

Tools are provided to administer and ease the upgrade process for Browser templates:

- A Browser checker utility, BrowserTemplateChecker.exe, introduced in 5.5 Service Pack 1 that is run after upgrade.
- Check the log file produced during the upgrade. In this log, detailed information can be found of the Browser templates which need to be adjusted to run correctly on Agresso Business World 5.6.x. To fix the problem, open the browser template, make the necessary changes and save.

Indexes and duplicates

When running the database check step in the wizard, there might be indexes missing due to duplicates.

Indexes might be very important, and missing or wrong indexes can lead to very poor and slow performance.

Check the log files if there have been any errors while creating the indexes.

See log file ..Database Script\Upgrade\Log\logindex_<database>.log

Use any database tool to find the duplicates (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

Example on how to find the duplicates:

```
select distinct client, attribute_id from agldimension  
group by client, attribute_id  
having count(*) > 1  
order by client, attribute_id  
/
```

Delete/change so there are no duplicates in the table, and re-create the index.

Look up in the Appendix, Agresso Data Dictionary to see the correct index definition. Create the index using the preferred DBA Tool (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

User defined views

The table structure is changed from release to release. Ensure all user defined objects are changed according to the new structure (See Appendix Agresso Data Dictionary)

Change the definition and recreate the views

See log file ..Database Script\Upgrade\Log\logviews_<database>.log

Remove tables not longer in use

The script *drop.asp* in the *Scripts* directory will drop all old and temporary tables no longer used by the application.

Note: This script must be run when the upgrade has been completely verified, and you are sure that none of the old tables are needed for backup purposes.

Production Upgrade (after Test)

UPGRADE OVERVIEW - PRODUCTION

Process description

The upgrade activities presented here, starts when you have a (more or less) fully functional 5.6 installation in the test environment. In addition, the Production database must have been upgraded to a 5.6 database. For database upgrades, see Upgrade Process Overview.

As for test, the upgrade process goes through four stages:

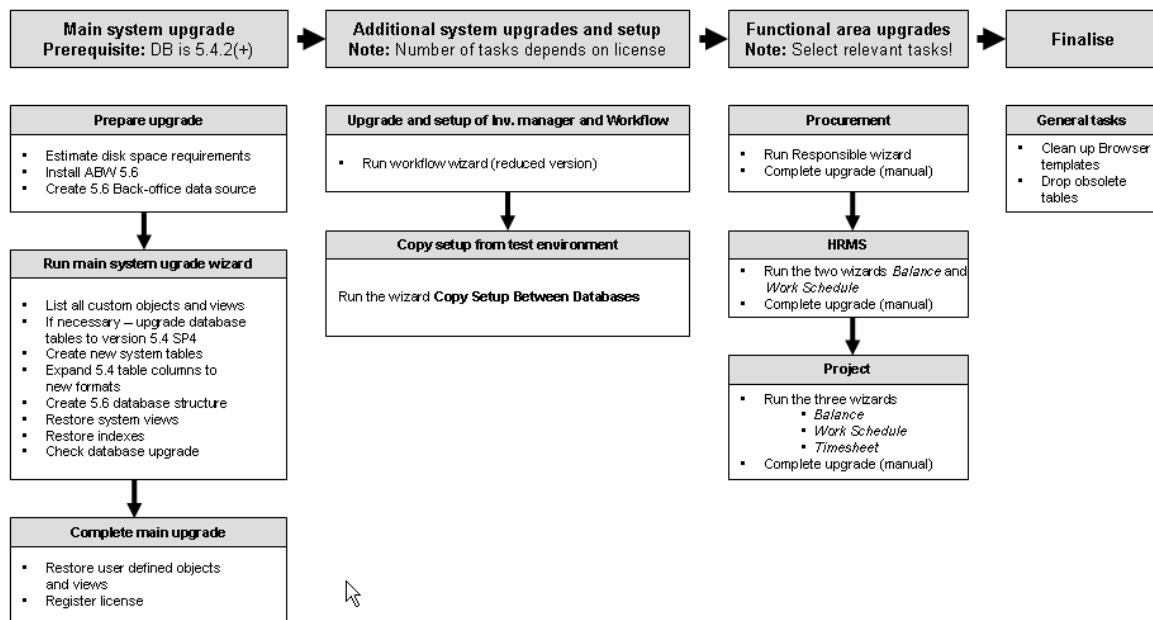
1. The Main system upgrade wizard performs all the required, basic upgrades of your system: It will add new tables, and convert old tables and data to the new formats - as far as it goes.
2. If relevant, you will run the Workflow wizard, with reduced functionality. And instead of manually establish roles and access rights, window options, data control etc. (as you did for the test environment), you will now run the wizard *Copy Setup Between Databases*. This will ensure that the

manual setup from the test environment is copied - as is - to production.

3. You must run upgrade wizards and perform manual setup for a few functional areas (or modules), where new implementations in ABW 5.6 prevent automatic upgrading. Modules not in use can be ignored.
4. Finally, you finish the upgrade by cleaning up browser templates and remove tables not longer in use.

Overview diagram

The diagram below summarises all tasks required for a complete upgrade - intended for the Production environment:



Main System Upgrade

PREPARE THE UPGRADE

Install 5.6 software

When your Test database is in the correct 5.4 version, you should install all 5.6 software.

Verify disk space

Available disk space needed for the upgrade is approximately the size of the largest table + 10%

Note: Expanding the maximum number of characters allowed in a field does not have any immediate impact on the database size. The expanded columns were previously of data type char, while they now are redefined to varchar.

Create 5.6 Data Source and initialise Business Server environment

Use the **Agresso Management Console** to create a new 5.6 Data Source connected to the old database.

When the connection is up and working, you must also initialise the Business Server environment (select the **Business Server** node in AMC and then **Initialise Business Server**).

Rename or delete amendment tables

Logging

If amendment logging is turned on for an Agresso table, an amendment table (or shadow table) are created and continuously updated with all table changes.

Note: During upgrade, all amendment logging will be turned off (by the wizard), and must be manually reset by using the ABW Smart client.

Table for amendment tables

All amendment tables are defined in the table [aagamendlog](#).

Naming standard: An Agresso table name is constructed from the structure `a<module><identification>` (example: `a cr client = acrclient`) while an amendment table is extended with the letters `shd` between `<module>` and `<identification>`.

If you turn on amendment logging for `acrclient`, the amendment table `acrshdclient` will be created and added to [aagamendlog](#).

Old amendment tables

During previous releases, the amendment tables have not been upgraded. When upgrading to 5.6, these tables will be checked, which may lead to a large number of errors.

Recommendation

To avoid problems with existing amendment tables, we recommend that you rename (or delete) all amendment tables before you start the upgrade process. Thereby, the upgrade wizard will not find them, and they will consequently not generate any errors.

If you need the amendment tables for historical reasons, they will always be available in your database copy.

To remember

Custom object definitions

The main system upgrade wizard will initially check the database for any custom objects, and write a detailed report containing all the old definitions (scripts). During the upgrade process, all custom objects will be removed.

The generated report, named [`user_def_obj.txt`](#), will be saved to the default log directory. You will need this report, as well as a detailed knowledge of the new table definitions (see Table changes in version 5.6) to restore previous functionality in Agresso 5.6.

The *custom objects* that will be removed, are:

- triggers and indexes,
- procedures,
- shadow tables,
- user defined views.

RUN THE UPGRADE WIZARD

The upgrade wizard

You find the upgrade wizard at the following location:

[...Agresso 5.6\DatabaseScript\DbUpgrade\UpgradeWizards.exe](#)

Upgrade scripts and log files

Logging and error handling

The **Log File** will, when the script has run, contain a description of what happened. Each statement are listed, along with status after execution.

Note: You should - as a general rule - always look at the log file when the script is finished. If any errors occurred, they must be corrected, and the script must be run again - *before* you continue with the next upgrade step.

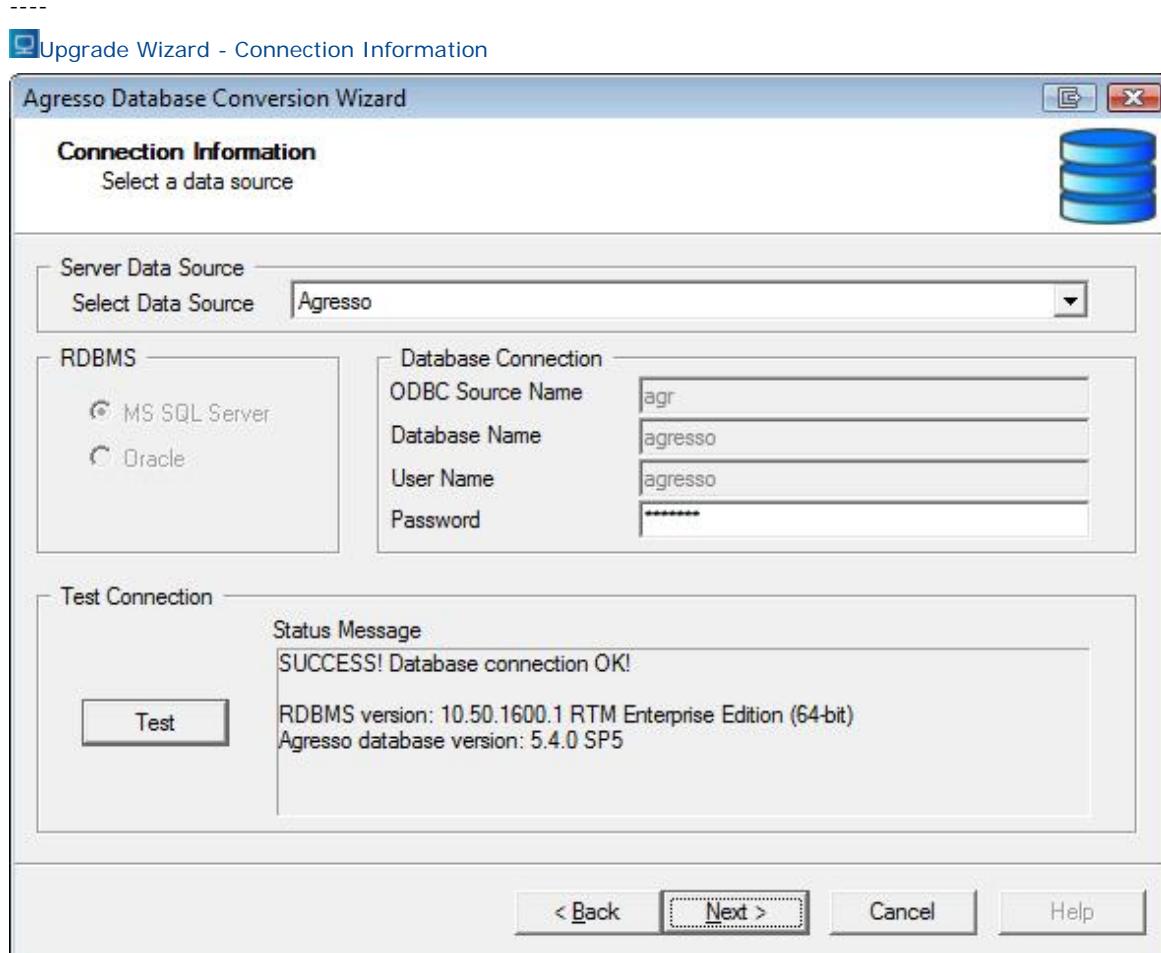
When we describe the actions for each step below, we *do not* ask you to view the log files. We take it for granted that you will do so!

Logging details

You can define the level of details to be listed in the log file by using the **Advanced** button available in the various wizard dialogs.

Run the upgrade wizard

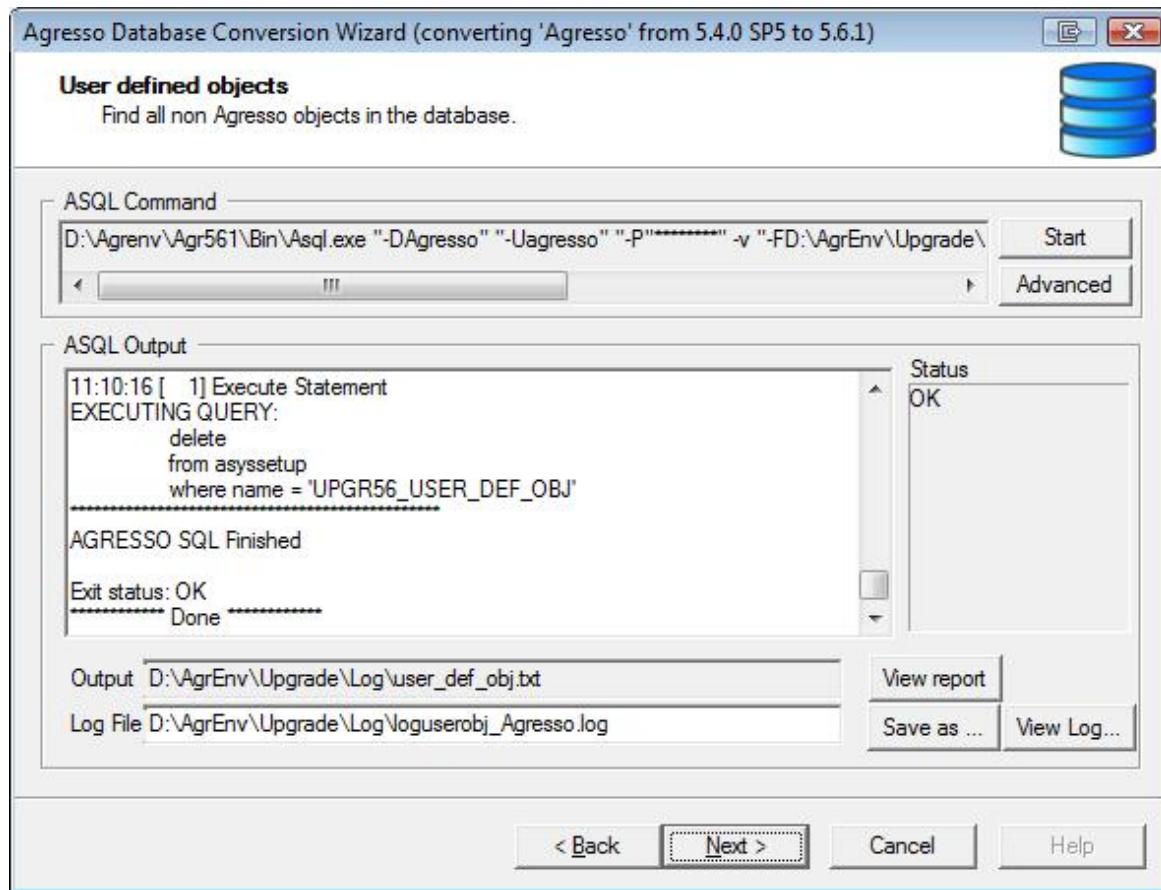
- When the upgrade wizard is up and running, select **Convert from 5.4**. Then select Step 1 *Main Upgrade Wizard*. The **Connection Information** dialog will be displayed.



- Select the data source, enter password, and click **Next**.

Note: If you get an error message, just accept the proposed action - and make a note of it! The error is not significant for the upgrade, but it will be when running Agresso later.





More step 3 details

Note: The report containing the custom object definitions are shown in the **ASQL Output** field. You will need this later.

Important: Do not continue if any errors are found in the database. These may be:

- **Null values** - these must be fixed before you continue. Use copyms/copyora to copy the table with the NULL values out, and then into the database again.
- **Duplicates**. Do not continue with the upgrade if there are duplicates in any of the tables
 - `aaguser`,
 - `acruserinfo` or
 - `aaguserfunc`.

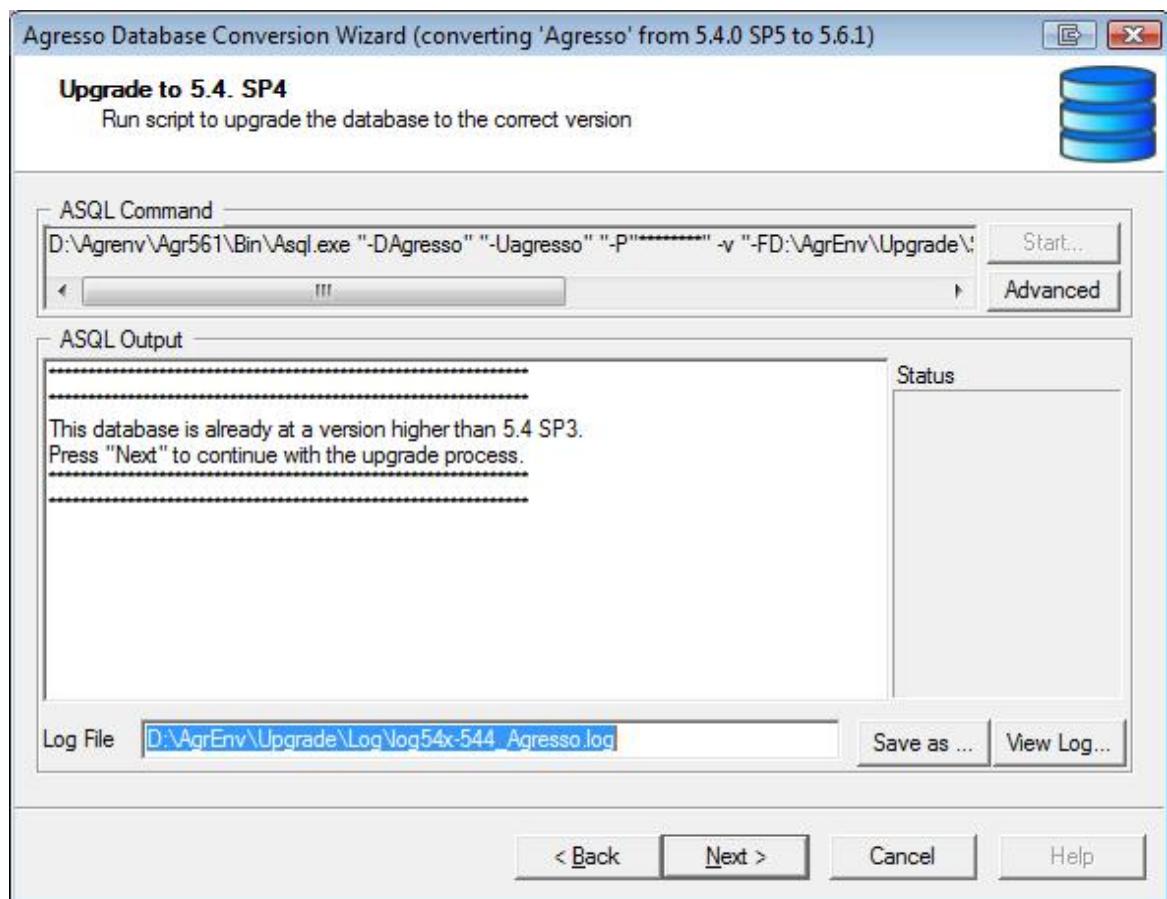
You must find the duplicates and remove them (or rename them).

Note: You must re-run the User defined objects step until there are no errors.

3. Click **Start** to run the script.

Important: You must correct all reported errors and make sure that everything is correct (re-run the script - until no errors are reported), before you can proceed to the next step.

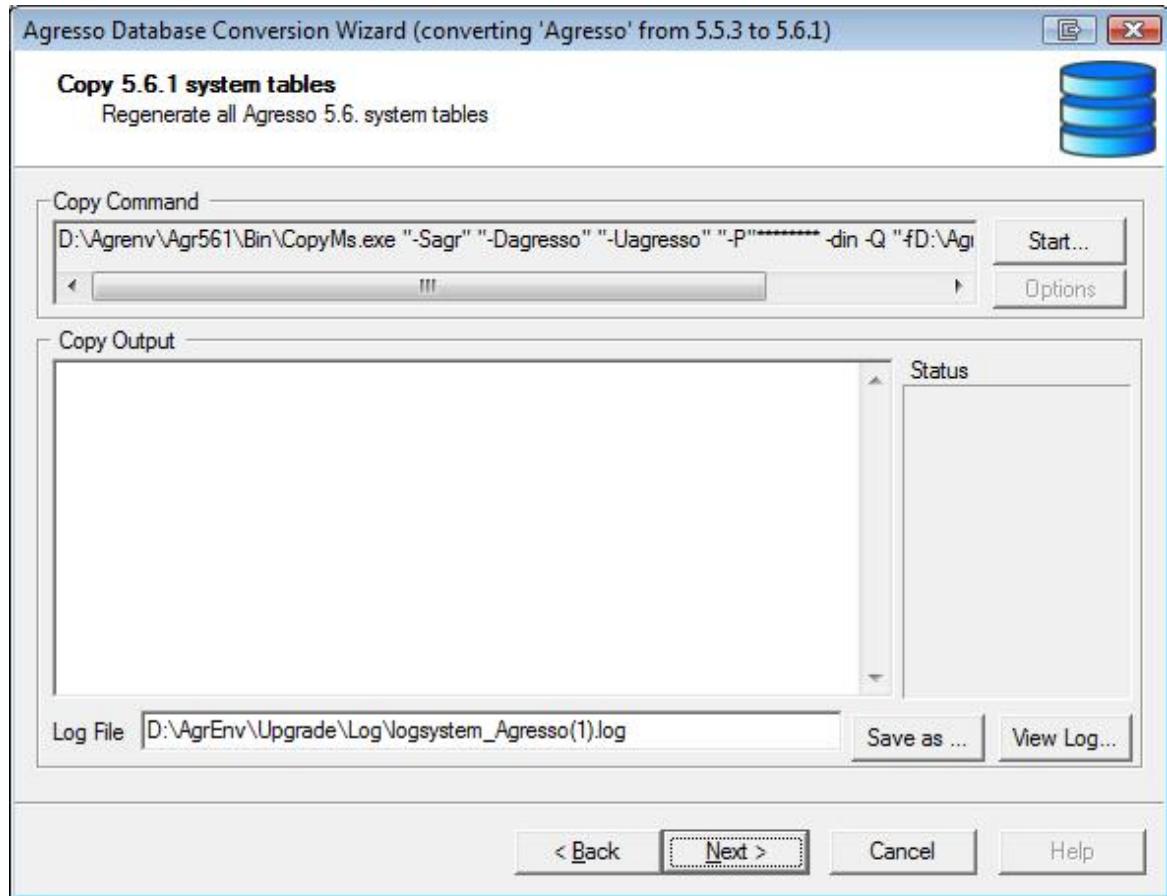
Upgrade Wizard - Upgrade to 5.4 SP4



4. If your database needs upgrading, click Start. Click Next to continue.

You are ready to restore the Agresso system tables, some awf* tables, and dictionary tables.

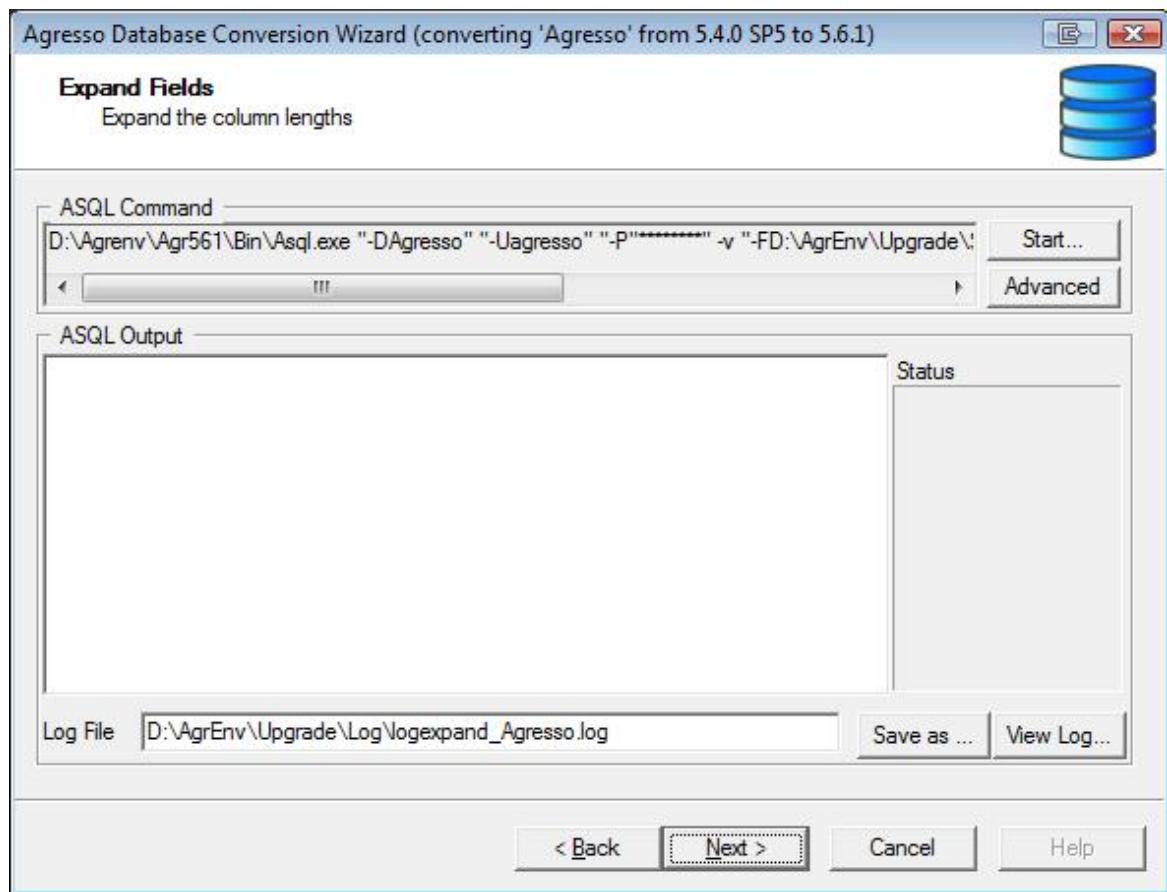
[Upgrade Wizard - Copy in System Tables](#)



5. Click **Start** to run the script, then **Next** to continue.

Next, the table columns will be expanded:

Upgrade Wizard - Expand Fields



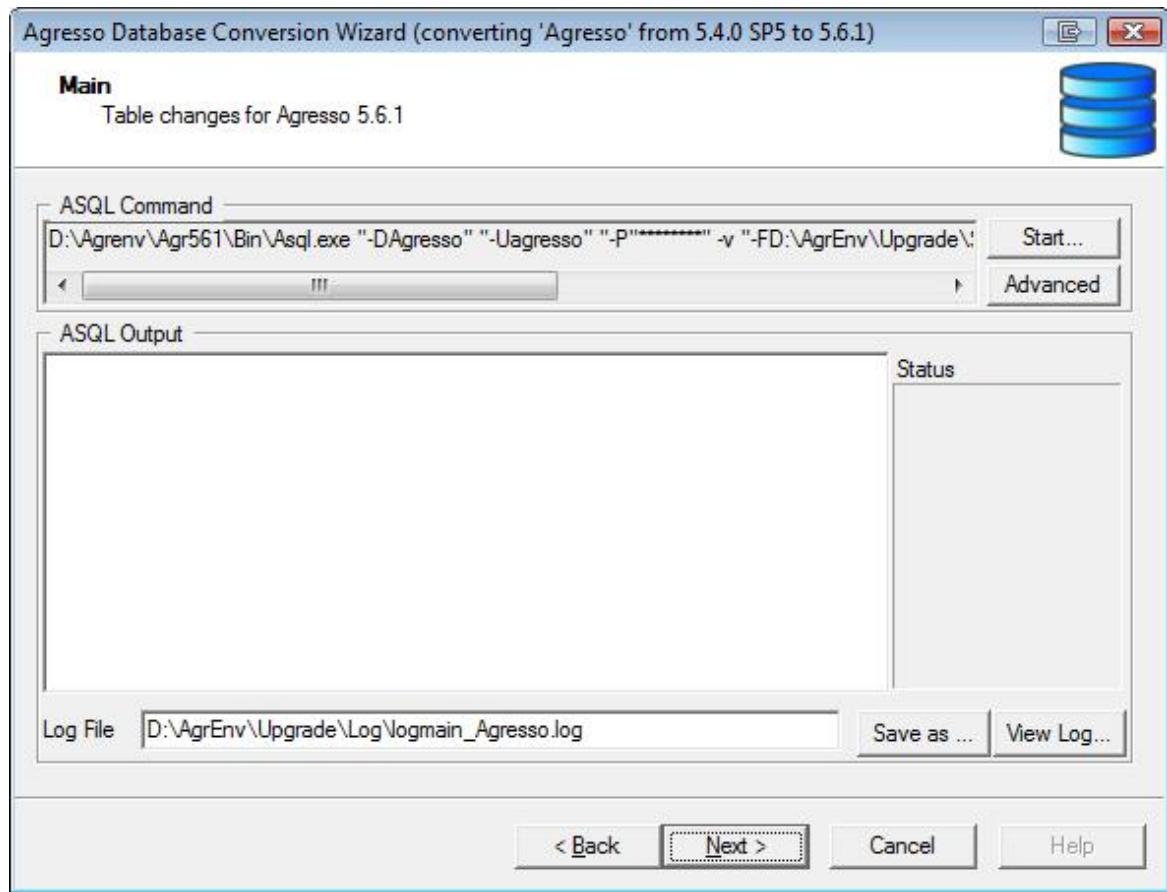
6. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

Note: This step might take some time.

Duplicates should be removed: If duplicates are found, it is recommended that the duplicates are removed before you continue. You do not have to rerun this step after the duplicates are removed.

You can now introduce the main changes in the database structure:

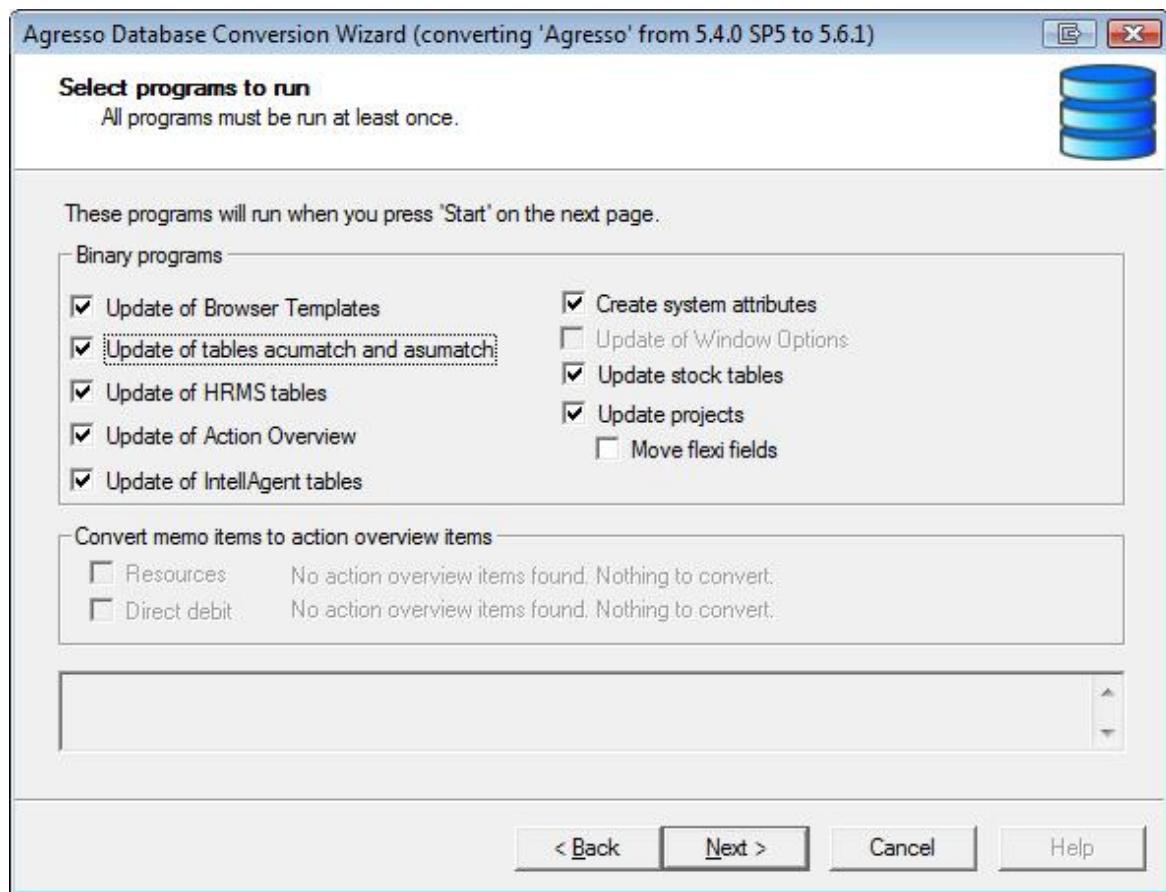
[Upgrade Wizard - Main - Table changes for Agresso 5.6](#)



7. Click **Start** to run the script, then (when the script is finished) click **Next** to continue.

The final changes to the database structure requires that you run some additional programs.

Upgrade Wizard- Run binary programs

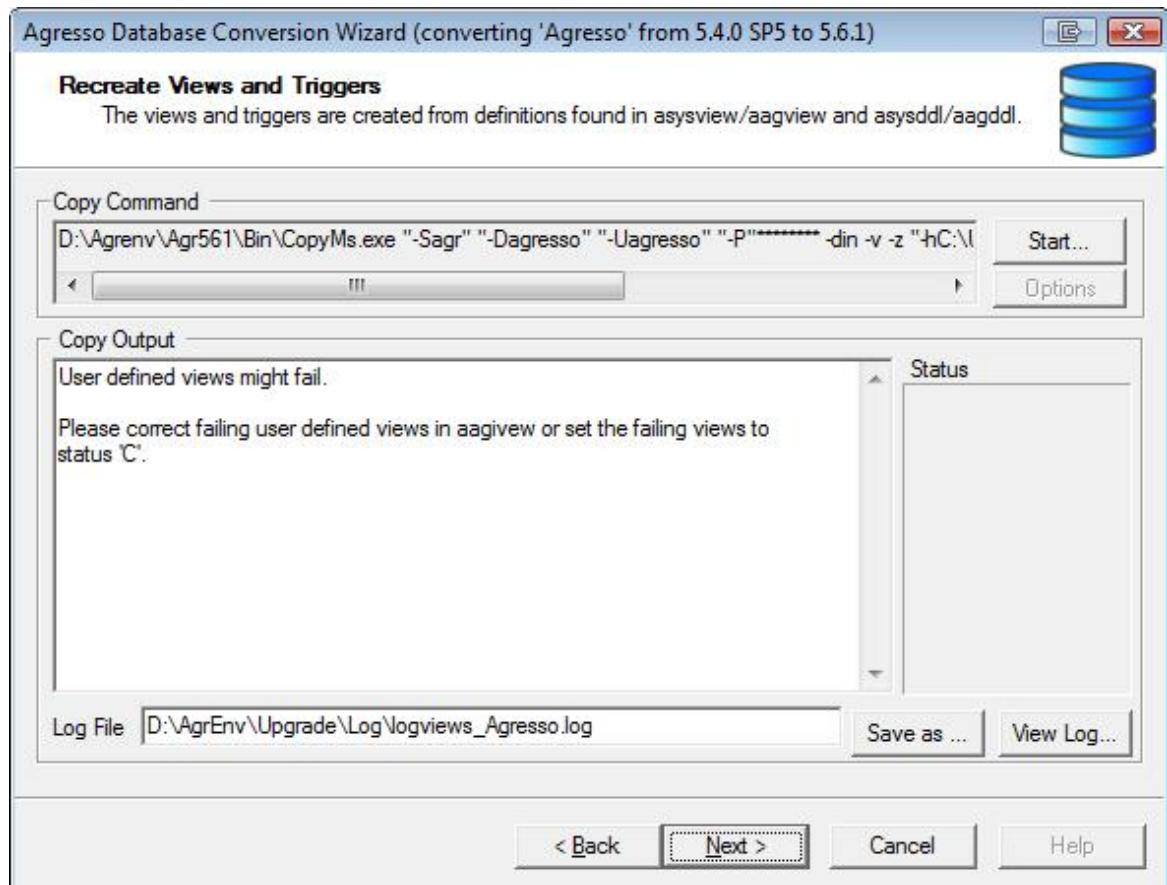


8. Select all programs listed and click **Next**.

Note: All programs must be run , but not necessarily at the same time. It will not do any harm if they are run more than once.

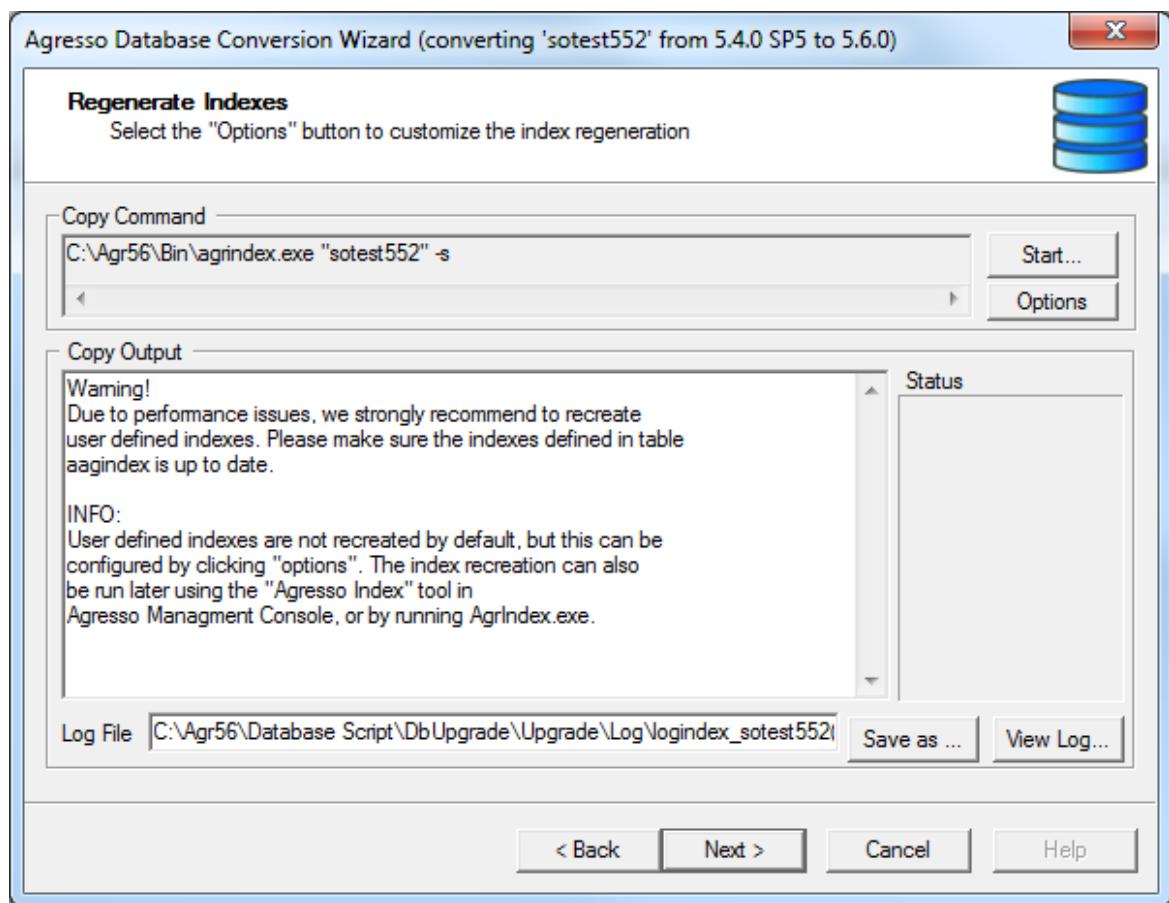
9. Click **Start** to run the selected upgrade programs, and then **Next** when the programs are completed.

 [Upgrade Wizard - Recreate Views and Triggers](#)

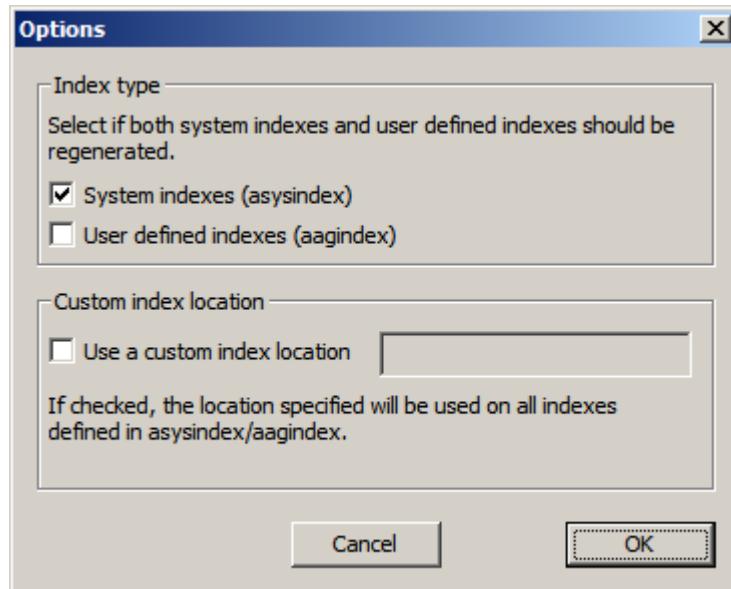


10. Click **Start** and **Next** in the next windows, in order to restore system views and indexes. This will bring you to the final step.

 [Upgrade Wizard - Regenerate Indexes](#)



11. Click the **Options** button to open the **Options** dialog:



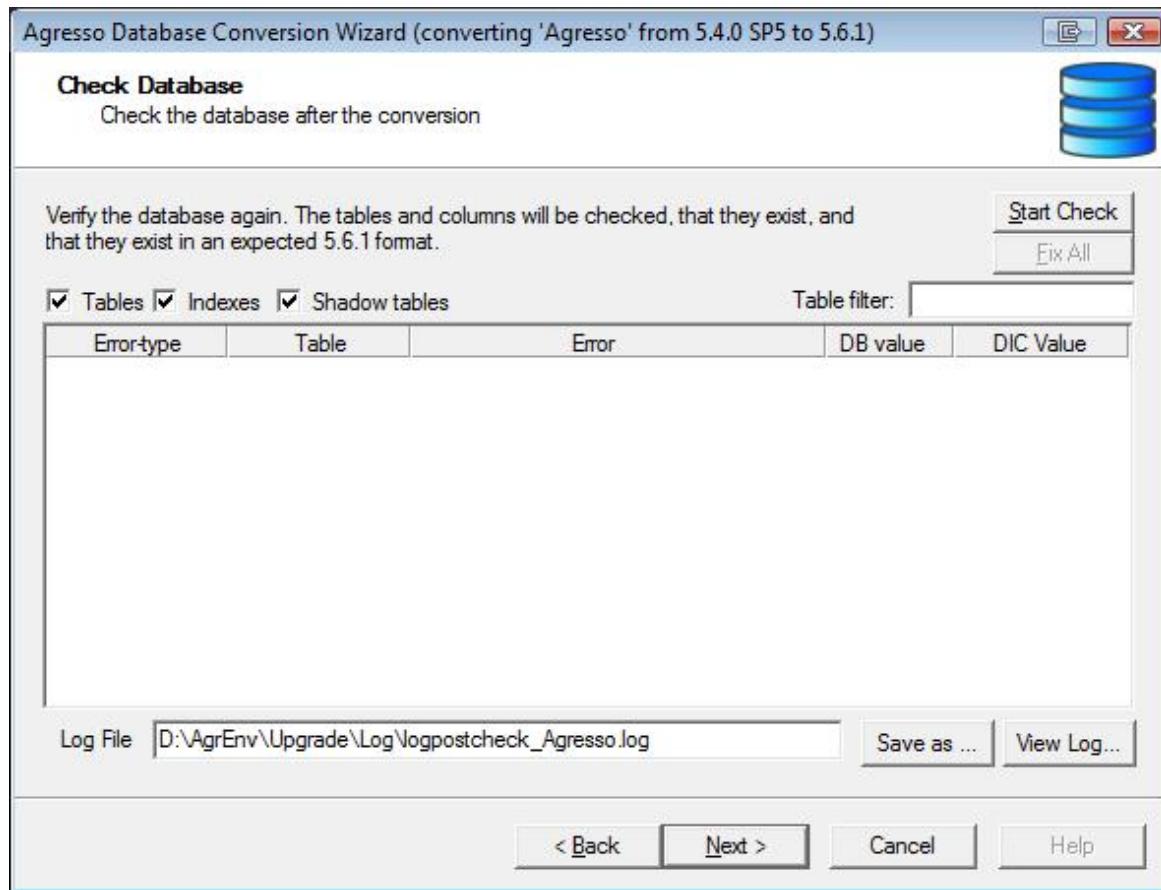
A note about indexes

By default, user defined indexes (stored in [aagindex](#)) are not re-created. Due to table changes, you should also change these, to avoid reduced performance. For details, see [AgrIndex](#).

12. Select options and click **OK**. Then click **Next >** to continue with next step (Check Database).

This check might take some time.

Upgrade Wizard - Check Database



13. Click Start Check.

This check compares the table structure, tables, columns, index with the format defined in Agresso's agrsys* tables.

If any differences occur, double-click on the error line to correct it. Or try the Fix All button. The Fix All might take some if tables with a lot of data needs to be fixed.

It is important to have the database with all the tables, columns and indexes in an expected format.

If there are tables with duplicates, clean up the duplicates and regenerate the index. A missing index might be crucial for the performance.

The following differences can be ignored:

Columns in the table are not found in the dictionary.

Columns are longer than expected.

COMPLETE THE MAIN UPGRADE

Prerequisites

The output file [*user_def_obj.txt*](#) will contain all necessary information for restoring user defined views and objects. It will also list all duplicate user Ids (Oracle only).

In order to restore previous, custom functionality, you will require [*user_def_obj.txt*](#), but also a detailed description of the 5.6 database structure. See relevant Table changes in the Appendix (Agresso Data Dictionary).

Tasks

To complete the main system upgrade, you should do the following:

1. Restore user defined objects and views

2. Register license

Restore user defined objects and views

A note about amendment logging

During the upgrade, all amendment logging are stopped.

Update tasks

The table below describes how custom objects can be upgraded:

Object type	Update tasks
Triggers and indexes	Use a database tool to change the columns and data types to fit to the new database structure.
Procedures	Use a database tool to re-define the procedures according to the new database structure.
Amendment logging (shadow tables)	Use the Smart client to open the AG30 Activation of logging server from the System Administration Data Control Amendment logging , and re-enable logging.
User defined views	Use the Smart client to open AG17 Database view definition . Zoom into each query, make the necessary changes, and remember to save the changes to have the views recreated.

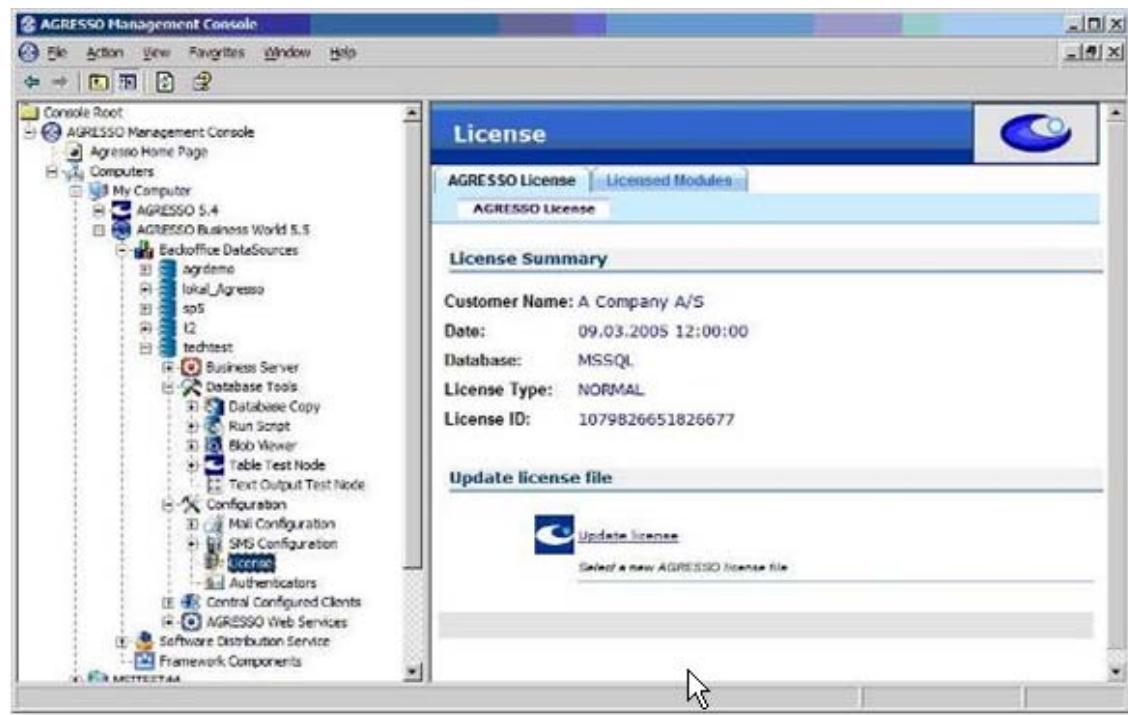
Register license

View licence

The license node in AMC allows you to add a new Agresso license to the database. It also enables you to view the licensed modules and the number of users registered for each module.

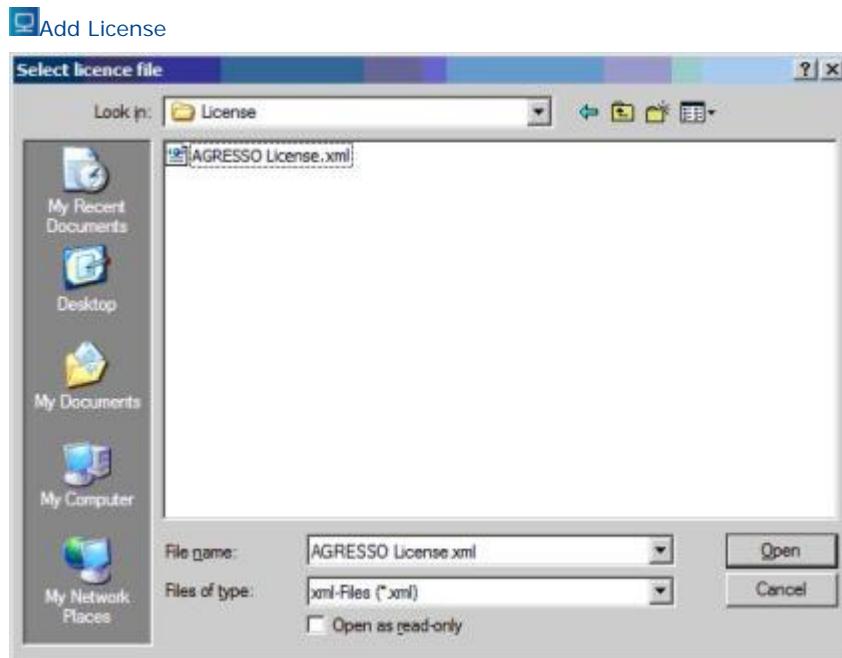
If you have a demo license installed, this node will inform you if the license will soon expire, or if it has already expired.

View License



Add License

To add a license you can either right-click on the license node and select **Add License** from the **All tasks** context menu. Or you can click on the [Add License](#) or [Update License](#) link found in the right pane of the management console. Then you will be prompted to select a license file.



You can also add a license using the Agresso Client Configuration tool.

Note: You need to enter a *new license* when upgrading from 5.4. to 5.6 to make all parts of Agresso work. If required, contact your local Agresso support office.

Login info

Your are now ready to test your Agresso Business World application with the following credentials (Note: **upgr55** is correct name when upgrading to 5.6):

Username:	upgr55
Client:	<your client>
Password:	upgr55

Additional System Upgrades

You can now proceed with Additional system upgrades and setup.

Additional System Upgrades and Setup

INVOICE MANAGER AND WORKFLOW

Agresso Workflow

New workflow solution

The workflow module (introduced in version 5.5 of ABW) replaces the Compello system. The new workflow module is based on an entirely new data model. Some Compello tables are kept to show historic workflow maps and are renamed to `ac<compello table name>`.

Wizard parameters

The Workflow upgrade wizard converts the data in the Compello tables to the new data model and defines processes in the new workflow system to match the existing set up in Compello.

Note: The wizard also makes it possible to upgrade rules from Compello. This is, however, NOT recommended, as the new workflow solution allows you to create far more advanced and generic rules than before.

Parameters

During the upgrade, you will be prompted to fill in some necessary parameters (see 3. below). These are explained as follows:

Parameter	Default value	Description
Client	*	The Agresso client to upgrade. Wildcards can be used (for example: *)..
Run as user		The user that will run the upgrade wizard.
Date From	19000101	Format: YYYYMMDD. Older document, transactions and images will be ignored.
Date To	20990101	Format: YYYYMMDD. Newer documents, transactions and images will be ignored.
Convert Invoice Manager registers	Off	Select if fixed registers from Compello shall be upgraded. If checked: <ul style="list-style-type: none"> • Users – Users will be created and the workflow flag will be checked for them. • Groups – will be converted to Roles • Substitutes and supervisor definitions
Convert Invoice Manager rules	Off	Only selectable if the Convert Invoice Manager registers is selected or the registers were upgraded in a previous run of the wizard. If selected, all Compello rules wil be upgraded to the new rules tables. Note: This is not recommended (see The upgrade wizard above).
Invoice Manager maps	Off	If selected, historic transactions and Compello map data will still be available. The old maps can be viewed from the new map viewer.
Create template workflow processes	Off	If selected, all template workflow processes will be copied to the clients chosen for upgrade. The template processes are drafts and will need to be committed by the user chosen in Run as user (see above) to become active. Note: This will overwrite any current workflow processes for the clients being upgraded.
Convert Invoice Manager archive	Off	If selected, Compello images will be upgraded to the new solution, and UNC-path (below) will be enabled.
UNC-path		UNC path to locate existing Compello documents. NB! Must be UNC path! Example: \\compelloarchive\\images

Upgrade Invoice manager

Upgrade consequences

The result of the upgrade processes described below is that

1. the Compello COM application is completely removed,
2. users will have direct access to the file share.

Dangers: The Compello document location must be configured as a file share (UNC name) with full access for everybody using the document archive. The share can be hidden, but there is no way of stopping an advanced user from mapping the drive and accessing the documents.

In practice, no one needs more than read-only access to Compello documents, and the documents are scrambled to prevent casual browsing.

Protection: We have tried combine these requirements, and reduce the security risks, by creating a dummy Windows user, which Agresso will use to access the documents. The system parameters DS_NATIVEFILE_USER and DS_NATIVEFILE_PWD allows you to configure this user. The dummy user (and no other) needs full access to the file share. When you run the document archive system within Agresso, the rights of this dummy user are picked up. Any attempt to map the file share outside of Agresso will be denied.

New document archive and old images

With the new document archive, customers with historical Compello images can select one of the following options:

- Leave existing documents in the Compello folders as read-only, for historical access only
- Move the documents into the Agresso Document archive (and remove them from the Compello database).

Required upgrade

To be able to view Compello documents in Agresso 5.6, you must at least:

1. Run the upgrade wizard to convert the database tables. (This can be done in stages by date etc.)
2. Configure permissions for the Compello file share.
3. Create a document type to access the Compello documents, for example Compello Invoices. This should belong to the GL Transaction key, and have the document system Compello selected.

This will change the pointers to the data only. The image files are not moved.

Complete migration

You may want to make a full migration for the following reasons:

- To have one document type only for all invoices, both old and new
- To be able to use the new Agresso document archive features

If you prefer a complete migration and remove the Compello archive, you can use the server process **DS01 Document copy** to move documents between document types. If the document types are based on different storage systems, as in this case, it automatically uses the drivers to do the conversion.

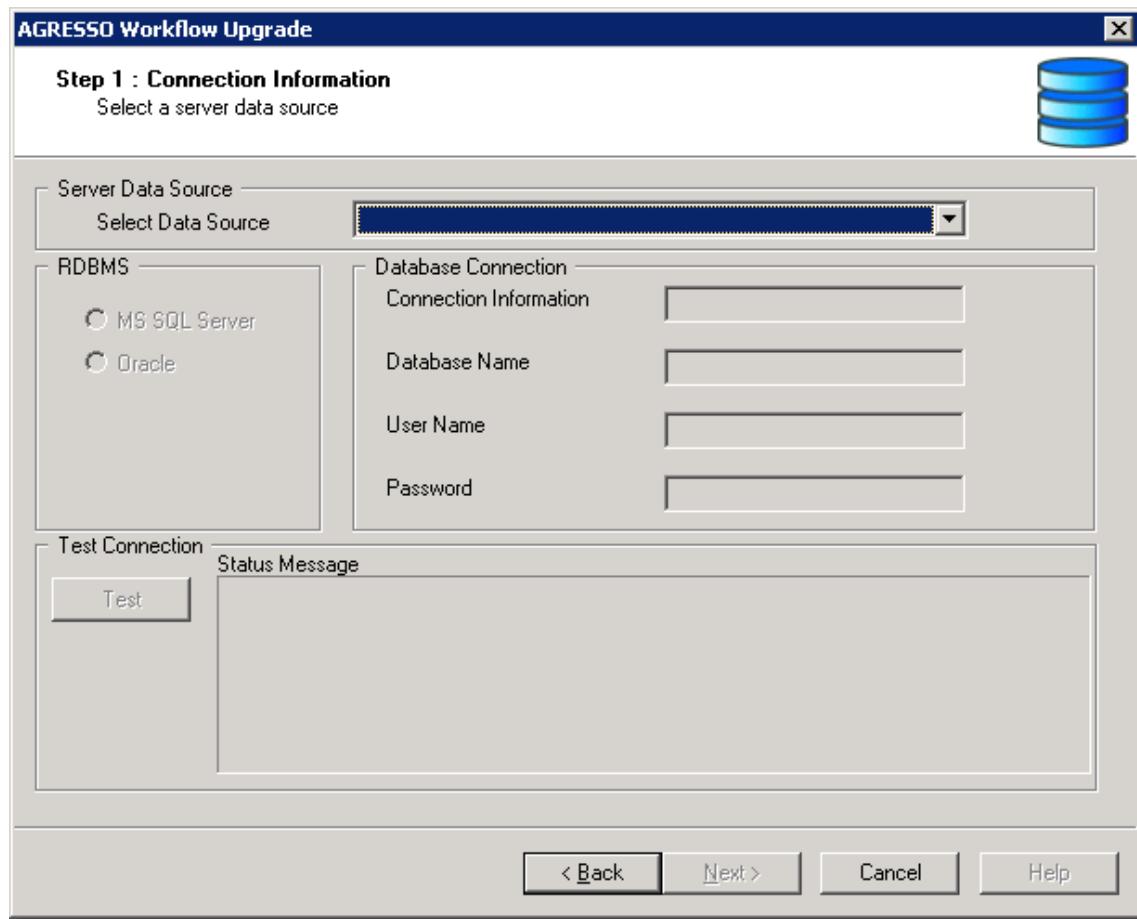
Run the wizard

Do as follows:

1. Start the upgrade wizard and move to the Step 1 dialog:



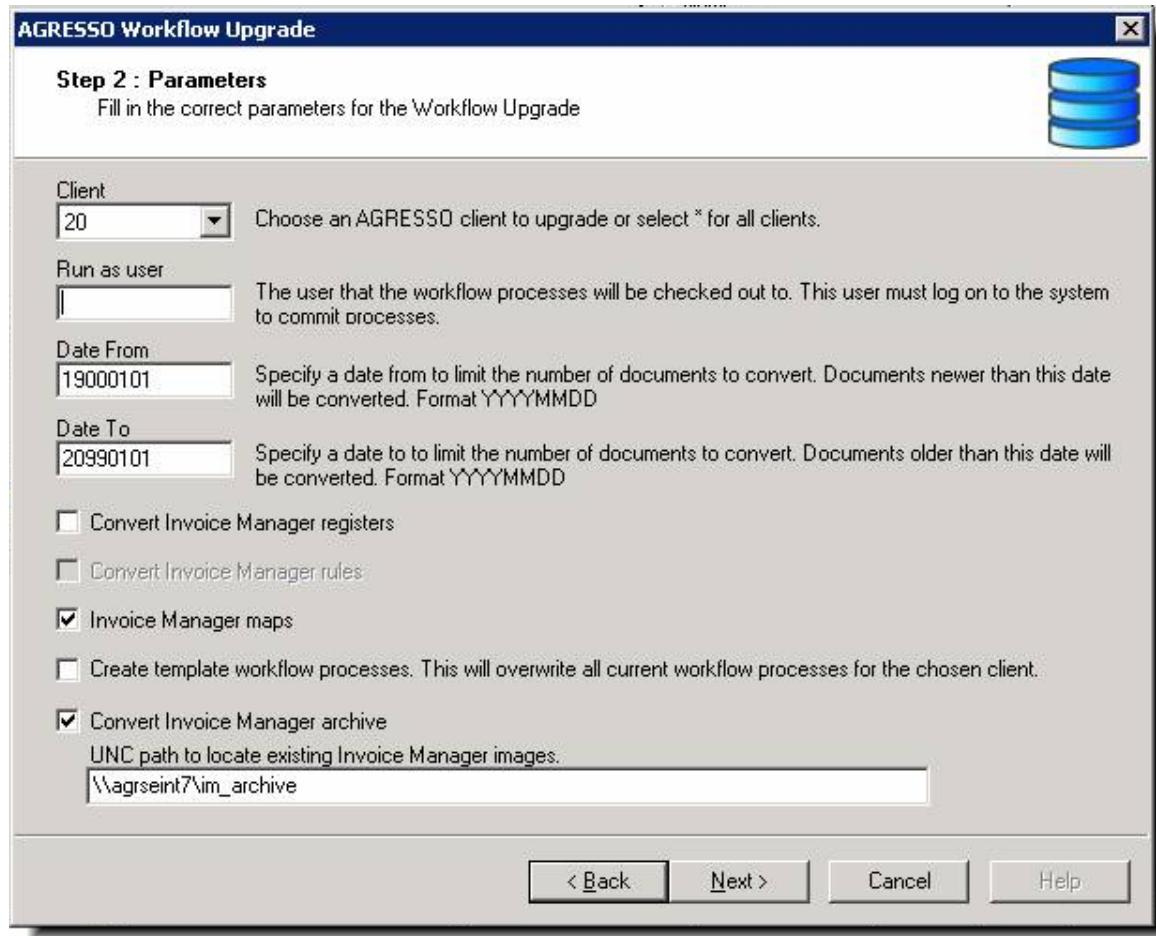
[Workflow Upgrade - Step 1](#)



2. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

You are now prompted to set necessary parameter values.

Parameters - Step 2



3. Make sure the parameter settings are correct, and click **Next**.

You are ready to run the upgrade scripts.

4. Click **Start** to run the script and the **Next** to continue.

The (basic) upgrade is completed and you can close the wizard.

DOCUMENT ARCHIVE UPGRADE

New Document archive solution

The Agresso Document archive was completely rewritten with ABW 5.5. The document archive is now an integrated part of a series of Agresso windows, and documents of any type (.doc, .pdf, .jpg etc.) can now be attached to most of the Agresso object's.

These profound changes require that all the previous documents must be converted.

Please refer to *Agresso 5.5 Release Notes, Document Archive* for details about technical and functional aspects of the new solution.

Invoice Manager Reference

See the topic [Workflow and Invoice Manager](#) for information on how to interface your old Compello images with the new Agresso document archive.

Running the upgrade wizard

Necessary preparations

During the upgrade, you will need to make a few selections in order to proceed (see 3. below). The parameters you must enter are explained in the following table:

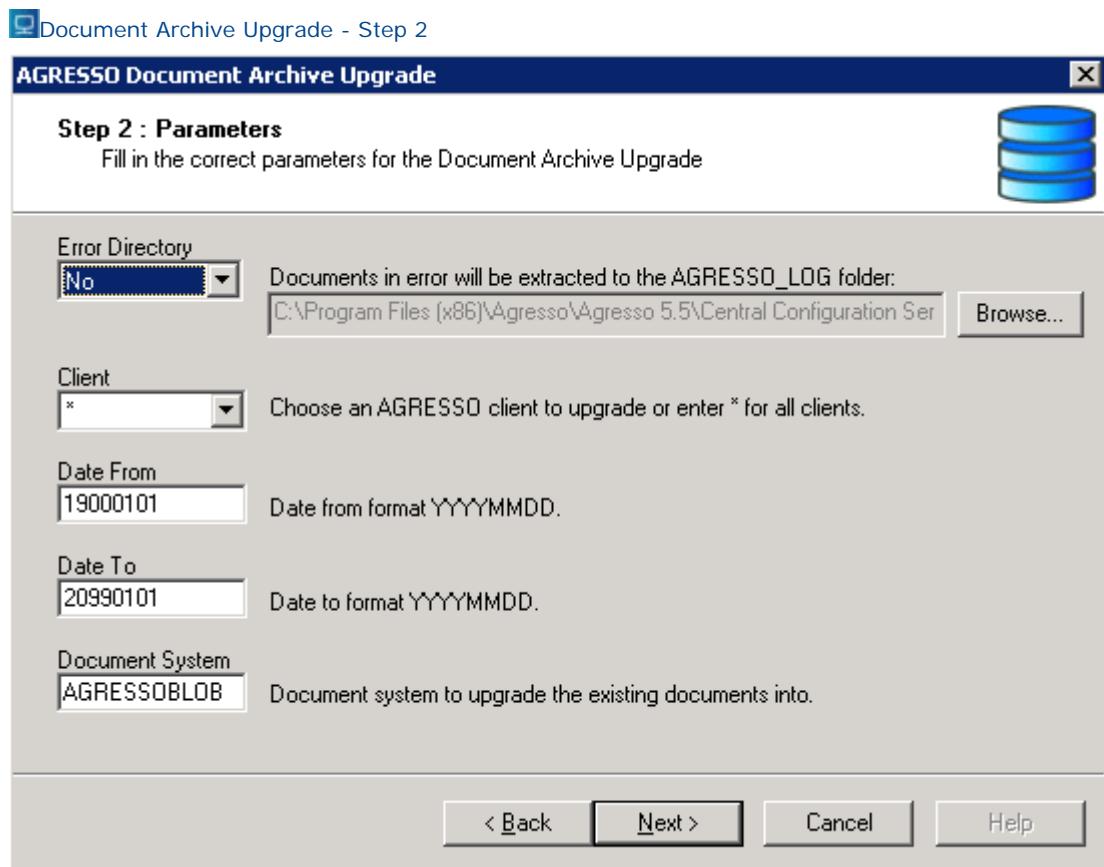
Field	Default value	Description
Error	No	Yes or No.

Directory		If Yes , erroneous documents will be extracted to the server logging (AGRESSO_LOG) directory for manual fixing. If No erroneous documents will just be reported in the log (not extracted).
Client	*	The Agresso client to upgrade. An asterisk (*) means all clients.
Date from	19000101	All documents registered at or after Date from will be included. Older documents will be ignored. Format: YYYYMMDD.
Date to	20990101	All documents registered before or at Date to will be included. Newer documents will be ignored. Format: YYYYMMDD.
Document system	AgressoBLOB	The new document archive can be connected to many document systems, including third party archives. You must choose a document system to upgrade the existing documents into. The default value is the new Agresso database archive.

Upgrade procedure

To upgrade to the new document archive solution, you will first start the Upgrade wizard. Continue as follows:

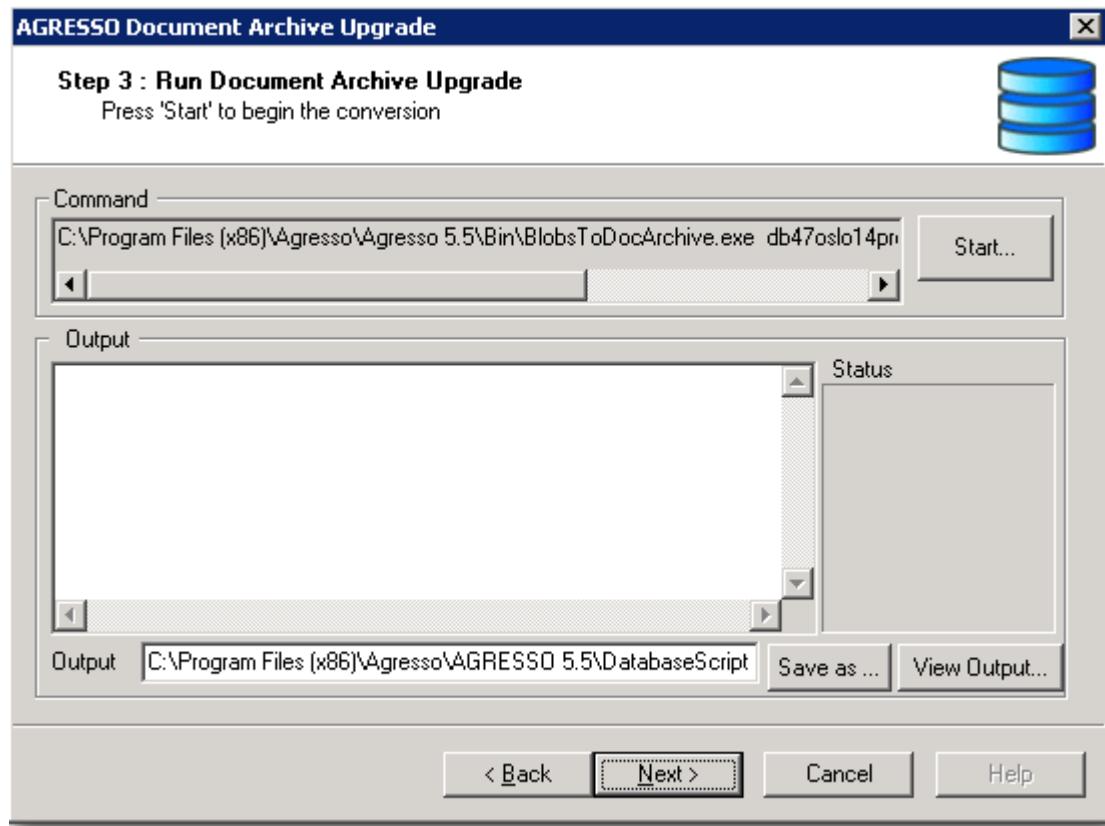
1. Double-click the [Document Archive](#) link in the Database Upgrade Wizards screen. After reading the introduction, click **Next**.
2. Enter connection information and click **Next** to prepare for step 2.



3. Fill in the required parameters and click **Next**.

You will now have to verify the path to the log file (output) made by the upgrade scripts.





4. Make sure that the output path is correct, and run the upgrade by clicking the **Start** button.

5. When the conversion is done, do as follows:

- Click **View Output** to study the conversion results
(we presume everything is in order)
- Click **Next** to bring up the final window.

6. Click **Finish**.

Note: If necessary, check the log file once more and make sure everything is correct.

MANUAL SETUP TASKS - PRODUCTION

The following areas require manual setup:

- Data control - see Release Notes for Common and System Administration, ABW 5.5 and 5.5.1 combined.
- Management of open items and Action overview - see Release Notes for Action Overview, ABW 5.5 and 5.5.1 combined

COPY SETUP BETWEEN DATABASES

Not to be used when upgrading to ABW 5.6!

Please note that the procedures described below is currently not valid! Send a request to the R&D forum.

General description

The **Copy Setup Between Databases** wizard copies the Agresso setup from one database to another. The following areas may be copied:

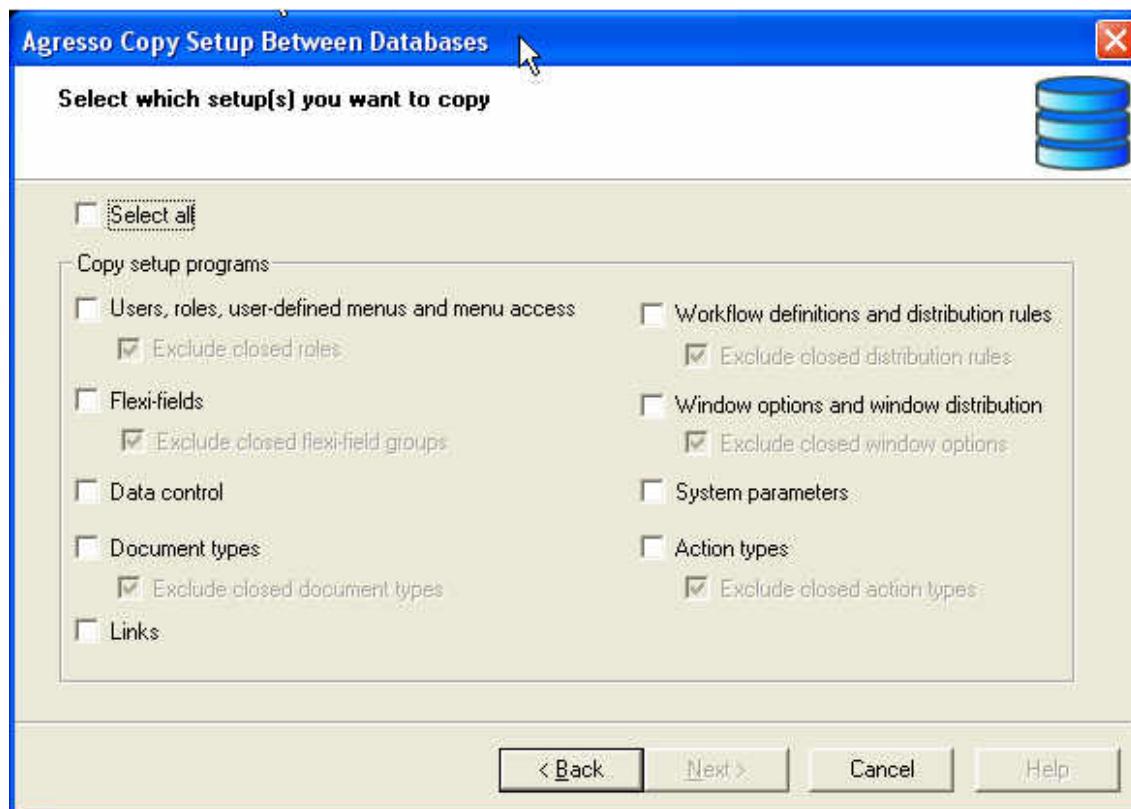
- Users, roles and menu access
- Workflow definitions and distribution rules

- Flexi-fields
- Window options and distribution of window options
- Data control
- System parameters
- Document types
- Action types

Note: Existing setup in the target database will be overwritten. See Dialog example below.

Dialog Example

The various options are presented in a single window:



Intended use

The wizard is created to facilitate transfer of a new and working configuration in the test environment, to the production environment.

Used in this context, it is important that all other updates are completed – on the test database – before you run the **Copy Setup Between Databases** wizard.

Run the wizard

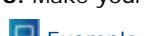
Note: If you need more details before you run the wizard, see [Wizard details](#) and [Updated tables](#) below

When the wizard is up and running, do as follows:

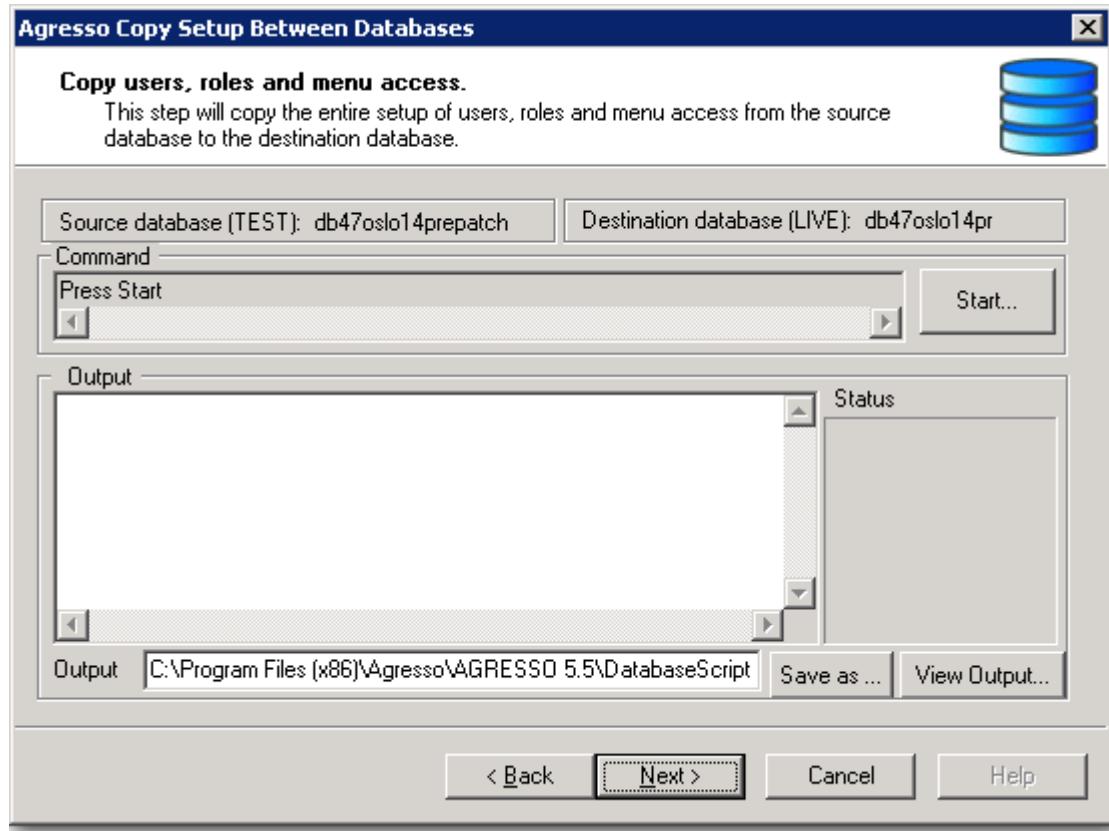
1. Enter connection information for the *Source* database and click **Next**.
2. Enter connection information for the *Destination* database and click **Next**.

You are presented with a few copy options. See Dialog example above.

3. Make your selections and click **Next**. The wizard is ready to start the Copy operation.



Example.



4. Click **Start**. You will need to confirm copying for each selected area.

Wizard details

The various copy programs (wizards) are described below:

The wizard ...	Will copy the following from source (test) database to target (production) database ...
Users, roles, user-defined menus and menu access	<ul style="list-style-type: none"> • New users (aaguser) with status Active • User information (addresses and passwords) connected to new users with status Active • All user detail information (aaguserdetail) • All roles • All menu access (aagaccess) granted to both users and roles • All user-defined menus (aagmenu) <p>You can exclude roles with status Closed from being copied to the destination database (default).</p>
Workflow definitions and distribution rules	<p>The entire setup of workflow and distribution rules. Existing setup will be replaced.</p> <p>You can exclude closed distribution (default).</p>
Flexi-fields	<p>The entire setup of Flexi-fields. Existing setup will be replaced.</p> <p>You can exclude closed Flexi-field groups (default).</p> <p>Note: No Flexi-field data will be copied!</p>
Window options and window distribution	<p>The setup of window options and distribution of window options. All existing setup will be replaced.</p> <p>You can exclude closed window options (default).</p>
Data control	<ul style="list-style-type: none"> • The entire setup of data control. All existing setup will be replaced.

	<ul style="list-style-type: none"> • All relations and relation values connected to attribute A11 –ROLEID • All setup made in Data control management (CR49). <p>Note: The program also updates all attributes with data control activated in the <i>source</i> database.</p>
System parameters	<p>System parameters.</p> <p>All changed parameters will be listed in a report file. The path to this report file is specified in the program's log file.</p>
Document types	<p>The setup of Document archive. All existing setup will be replaced.</p> <p>Note: No documents will be copied!</p>
Action types	<p>The setup of Action overview. All existing setup will be replaced.</p>
Links	<p>The setup of Links. All existing setup will be replaced.</p>

Updated tables

Below we list the tables affected by the copy programs, and indicate the operations they are exposed to:

Users, roles, user-defined menus and menu access

The following tables are affected:

Table name	Data replaced	Rows inserted	Rows deleted
awfelemttype	X		
awfelemtypedet	X		
awfelemtypelogrouping	X		
awfelemtypeitem	X		
awfelemtypemap	X		
awfprocaction	X		
awfprocedelines	X		
awfprocdelay	X		
awfprocdelaydet	X		
wawfprocelemtype	X		
awfprocess	X		
awfprocfunction	X		
awfprocrole	X		
awfprocsplit	X		
awfrule	X		
awfruledet	X		
awfrulegroup	X		
awfversion	X		
acrdiagramlayout	X		
awfalternate	X		
awfbmethods	X		

awfcolumns	X		
awfelemtypemenu	X		
awfmanstep	X		
awfprocrcd	X		
awfprocsin	X		
awfuserdetail	X		
aimblob		X	X

Flexi-fields

The following tables are affected:

Table name	Data replaced	Rows inserted	Rows deleted
acractionelemtype	X		
acractiotype	X		
agldimvalue		X	X
agldescription		X	X
acraktioncontact	X		
acraktionattval	X		
acracctionattvaldet	X		
awfelemtyp	X		
awfelemtypedet	X		
awfelemtypeitem	X		

Links

The following tables are affected:

RESPONSIBLES UPGRADE

Changes in data structure

The new solution for users and roles has required a completely new implementation of the *Responsible* concept in the Logistics module. Previously, the *algresponsible* table hold information about responsible codes and the *resource_id* (not the *user_id*) that was linked to the code.

Agresso 5.5 introduced a solution where a responsible belongs to a certain role, and where role membership is based on the *user_id*.

Upgrade of orders created by Agresso 5.4x

The main purpose of this Responsibles Upgrade, is to convert active orders (created in the various logistics modules) in such a way that the responsible person still can be identified when the order is further processed.

Note

The upgrade will *not* convert the old responsible resources into new responsible *roles*.

Upgrade tasks

The main upgrade tasks are performed by the responsible upgrade wizard, and are described as follows:

Task	Description
Automatic code matching	Match resource_ids from the algresponsible table with the 5.6 users. If a user can be uniquely identified by a resource_id, the upgrade wizard will establish a new connection between the 5.4 responsible code and the 5.5 user id.
Manual matching	For all unmatched resource_ids, you are prompted to manually enter user_ids. There is no requirement that the new user really takes part in 5.5 responsible role. This task can be repeated several times, until all (old) resource_ids have a valid match.
Automatic order upgrade	The wizard will search through all relevant orders and convert all orders with a responsible codes where there has been a successful match. Log: The result will be written to the log, and can be viewed in the table <i>algprophead</i> .
Manual order update	For all the orders that were not converted, you will have to use the registration windows for the various order types, to set the correct responsible. Note: This requires that the various responsible roles are set up in the Responsible Setup window in the Smart client. See <i>Agresso 5.5 Release Notes, Logistics – Common logistics</i> .

Run the upgrade wizard

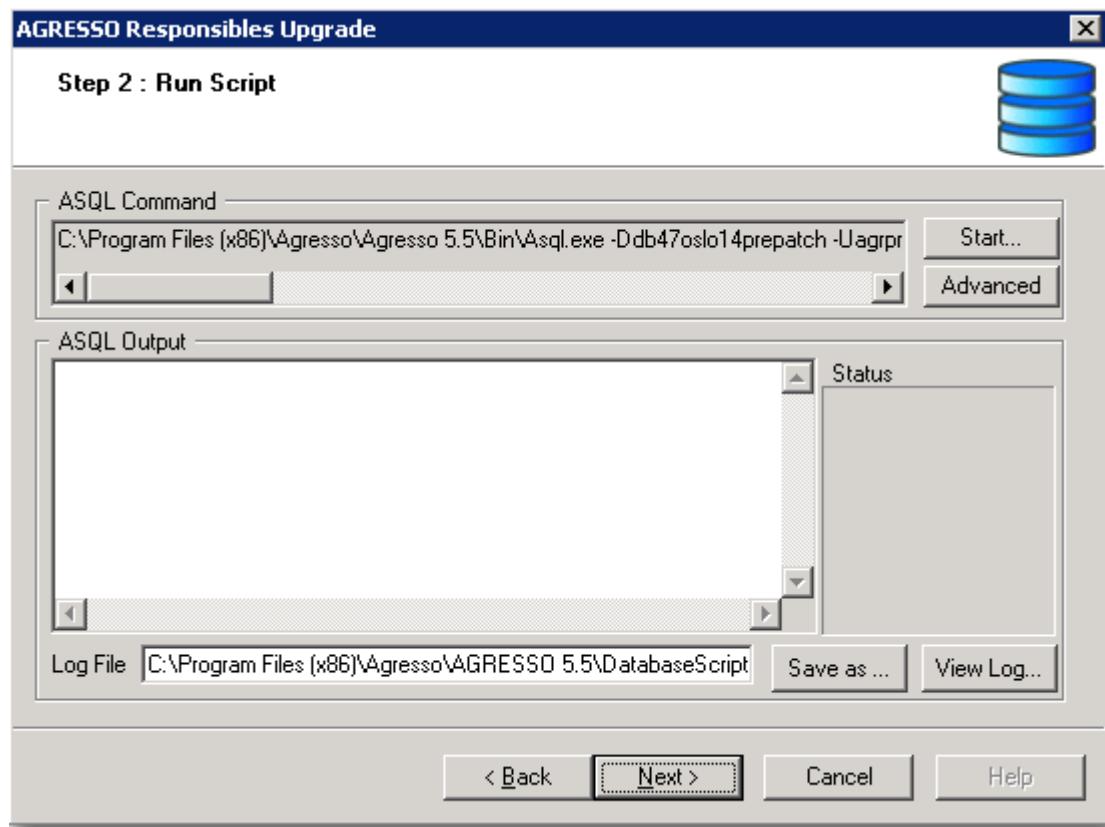
When the **Responsibles Upgrade wizard** is up and running, navigate to the Step 1 window, and do as follows:

1. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

You are now ready to run the upgrade scripts.

Note: If you already have run the upgrade scripts once, and in addition performed some manual matching (3. below), a re-run of the wizard will remove all manual updates!

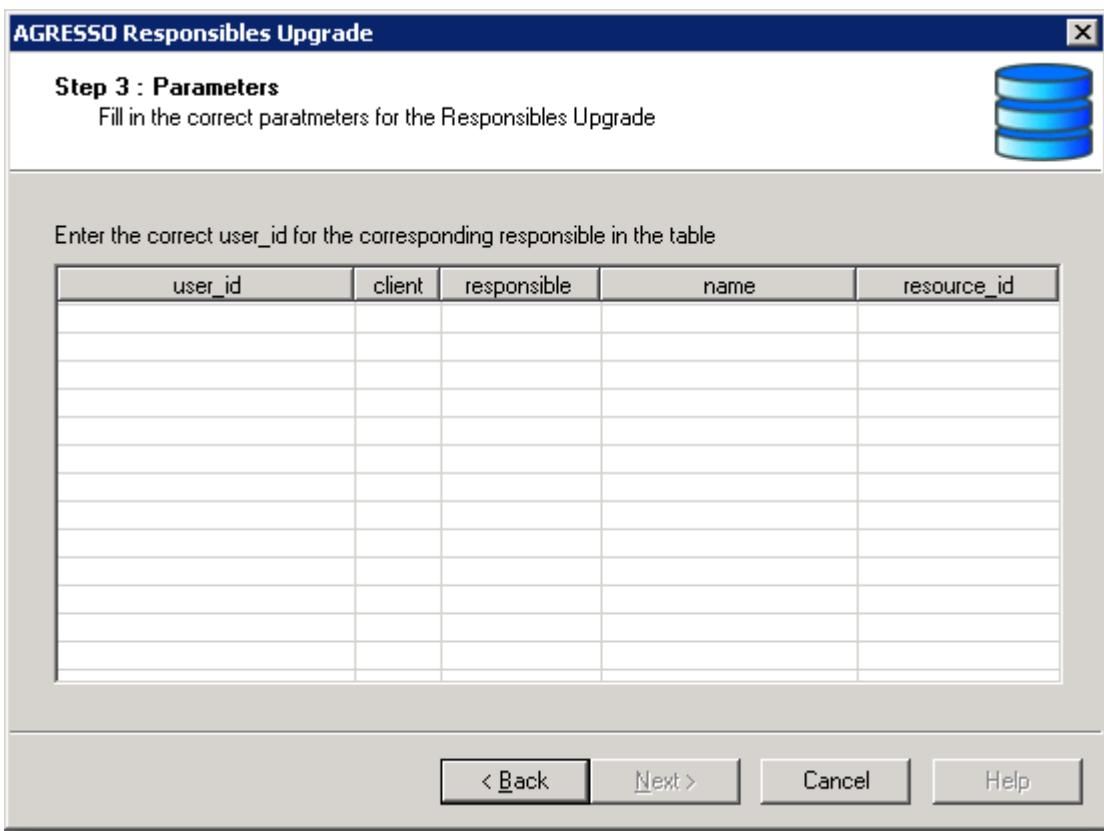
 **Responsibles Upgrade - Step 2**



2. Click **Start** to run the script, and then **Next** (we assume that everything went OK after **Start**). The wizard will match all identifiable resource_ids with the 5.5 user_id. When completed, the status for the *algresponsible* table is set to S.

You will now be prompted to manually enter missing user_ids:

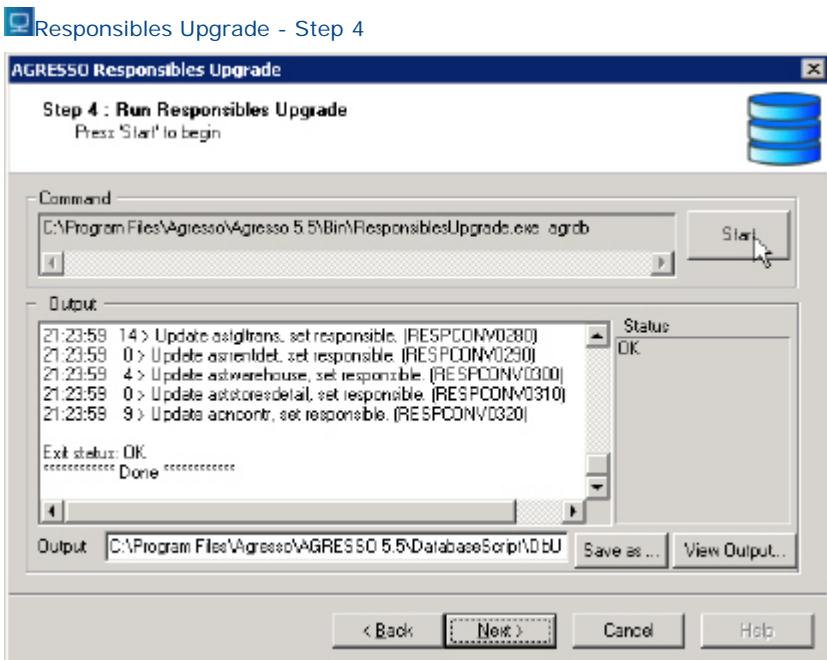
Responsibles Upgrade - Step 3



3. Repeat steps a. and b. as many times as necessary:

- a. Add missing user_ids.
- b. Click **Apply**
- c. Click **Next** when you are finished.

The next task is automatic order upgrade.



4. Click **Start**, and then **Next** to finish.

This wizard can be run several times if you would like to add user_id's at a later stage.

Complete upgrade - additional manual tasks

There will normally remain some unmatched responsible codes on the orders. If so, the responsible roles must be registered in the Agresso Smart Client, and the orders must be updated in their respective registration windows.

HRMS UPGRADES AND SETUP

Wizards

You must run two wizards:

- [Balance Upgrade](#) and
- [Work Schedule and Absence Details Upgrade](#)

Manual setup

The manual setup tasks are described in the following documents:

- *Release notes, AGRESSO 5.5, HRMS/PCB integration*
- *Service Pack notes, AGRESSO Business World 5.5 SP2, Project*

PROJECTS UPGRADE AND SETUP

Wizards

To upgrade the Projects module, you will need to run the following wizards:

- [Balance Upgrade](#) - if not already run,
- [Work Schedule and Absence Details Upgrade](#) - if not already run, and
- [Timesheet Upgrade](#).

Manual setup

See the documents:

- *ABW 5.5 Reference manual, PCB Setup resources, time costs & income*
- *Service Pack notes, AGRESSO Business World 5.5 SP2, Project*

Finalise

FINALISE UPGRADE

Overview

To complete the upgrade, you should:

- Check the converted Browser templates, and correct any errors.
- Clean up duplicates and add indexes.
- Correct all user defined views due to the table changes (see Appendix, Agresso Data Dictionary)
- Remove all tables no longer in use.

Check and correct Browser templates

Tools are provided to administer and ease the upgrade process for Browser templates:

- A Browser checker utility, BrowserTemplateChecker.exe, introduced in 5.5 Service Pack 1 that is run after upgrade.
- Check the log file produced during the upgrade. In this log, detailed information can be found of the Browser templates which need to be adjusted to run correctly on Agresso Business World 5.6.x. To fix the problem, open the browser template, make the necessary changes and save.

Indexes and duplicates

When running the database check step in the wizard, there might be indexes missing due to duplicates.

Indexes might be very important, and missing or wrong indexes can lead to very poor and slow performance.

Check the log files if there has been any errors while creating the indexes.

See log file ..Database Script\Upgrade\Log\logindex_<database>.log

Use any database tool to find the duplicates (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

Example on how to find the duplicates:

```
select distinct client, attribute_id from agldimension
group by client, attribute_id
having count(*) > 1
order by client, attribute_id
/
```

Delete/change so there are no duplicates in the table, and re-create the index.

Look up in the Appendix, Agresso Data Dictionary to see the correct index definition. Create the index using the preferred DBA Tool (for example Query Analyzer (SqlServer), Sqlplus (Oracle)).

User defined views

The table structure is changed from release to release. Ensure all user defined objects are changed according to the new structure (See Appendix Agresso Data Dictionary)

Change the definition and recreate the views

See log file ..Database Script\Upgrade\Log\logviews_<database>.log

Remove tables not longer in use

The script *drop.asp* in the *Scripts* directory will drop all old and temporary tables no more used by the application.

Note: This script must be run when the upgrade has been completely verified, and you are sure that none of the old tables are needed for backup purposes.

Wizard descriptions

DOCUMENT ARCHIVE UPGRADE

New Document archive solution

The Agresso Document archive was completely rewritten with ABW 5.5. The document archive is now an integrated part of a series of Agresso windows, and documents of any type (.doc, .pdf, .jpg etc.) can now be attached to most of the Agresso object's.

These profound changes require that all the previous documents must be converted.

Please refer to *Agresso 5.5 Release Notes, Document Archive* for details about technical and functional aspects of the new solution.

Invoice Manager Reference

See the topic [Workflow and Invoice Manager](#) for information on how to interface your old Compello images with the new Agresso document archive.

Running the upgrade wizard

Necessary preparations

During the upgrade, you will need to make a few selections in order to proceed (see 3. below). The parameters you must enter are explained in the following table:

Field	Default value	Description
Error Directory	No	<p>Yes or No.</p> <p>If Yes, erroneous documents will be extracted to the server logging (AGRESSO_LOG) directory for manual fixing.</p> <p>If No erroneous documents will just be reported in the log (not extracted).</p>
Client	*	The Agresso client to upgrade. An asterisk (*) means all clients.
Date from	19000101	All documents registered at or after Date from will be included. Older documents will be ignored. Format: YYYYMMDD.
Date to	20990101	All documents registered before or at Date to will be included. Newer documents will be ignored. Format: YYYYMMDD.
Document system	AgressoBLOB	The new document archive can be connected to many document systems, including third party archives. You must choose a document system to upgrade the existing documents into. The default value is the new Agresso database archive.

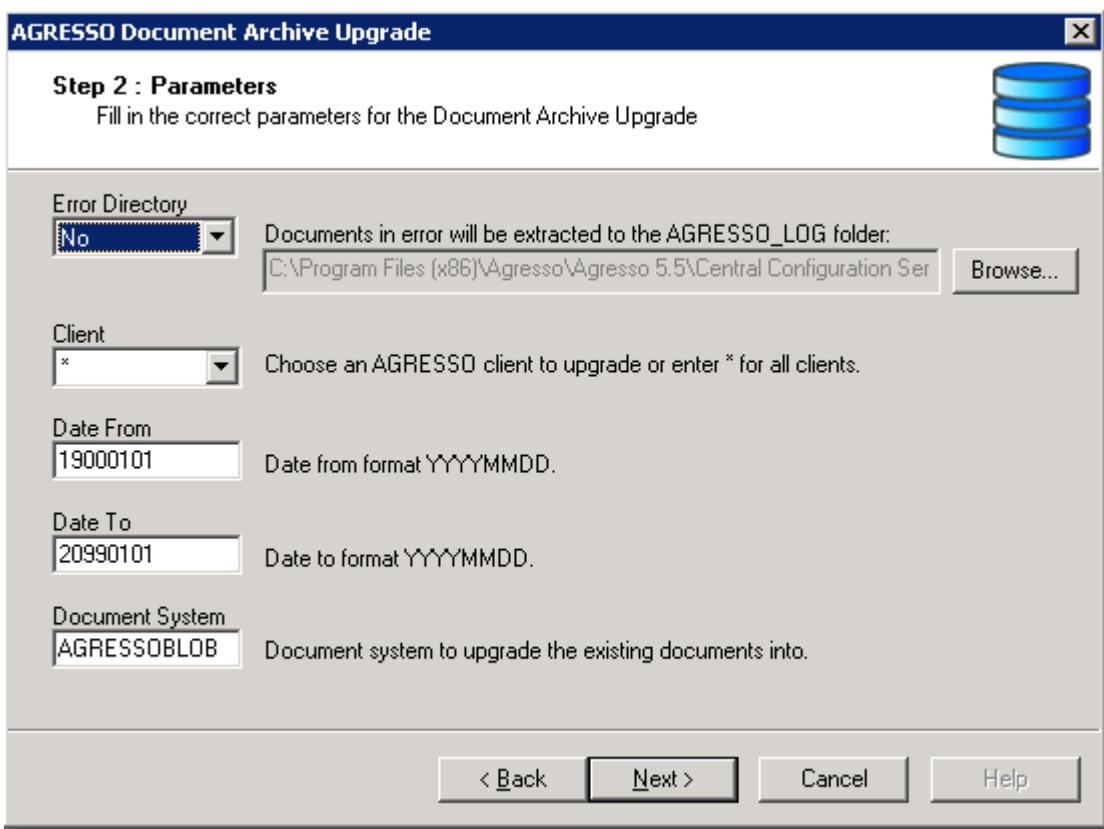
Upgrade procedure

To upgrade to the new document archive solution, you will first start the Upgrade wizard. Continue as follows:

1. Double-click the [Document Archive](#) link in the Database Upgrade Wizards screen. After reading the introduction, click **Next**.
2. Enter connection information and click **Next** to prepare for step 2.

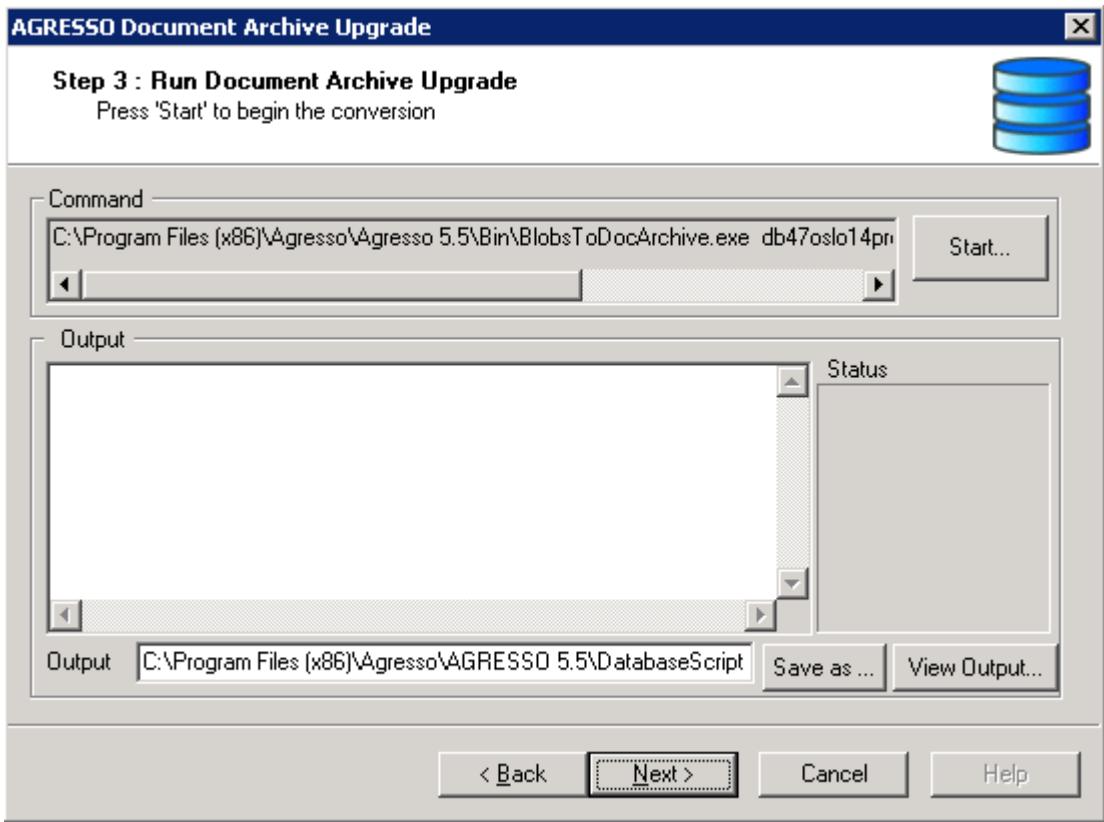


Document Archive Upgrade - Step 2



3. Fill in the required parameters and click **Next**.
You will now have to verify the path to the log file (output) made by the upgrade scripts.

Document Archive Upgrade - Step 3 #1



4. Make sure that the output path is correct, and run the upgrade by clicking the **Start** button.

5. When the conversion is done, do as follows:

- a. Click **View Output** to study the conversion results
(we presume everything is in order)
- b. Click **Next** to bring up the final window.

6. Click **Finish**

Note: If necessary, check the log file once more and make sure everything is correct.

CREATE SYSTEM ATTRIBUTES

Description

The wizard **Create System Attributes** will generate all the fixed attributes currently missing in your ABW 5.6 installation.

The wizard is part of Step 6 – **Run binary programs** in the main system upgrade wizard, but is also available in a stand alone version, available in the file *CreateAttributesWizard.exe*.

Standard (fixed) attributes only

The wizard updates only attributes delivered as part of the standard **Agresso Business World** installation (with attribute id from **A0** to **LZ**).

Other attribute types, for example localisation specific attributes or custom attributes, will not be updated by the wizard.

All clients will be updated

Attributes are client dependent and the wizard will generate one attribute instance per client. Attribute names and descriptions will comply with the selected language (Company Information screen, CR01) for the various clients.

Run the wizard

The wizard is very straightforward, with no complex options. Just follow the instructions on screen.

RESPONSIBLES UPGRADE

Changes in data structure

The new solution for users and roles has required a completely new implementation of the *Responsible* concept in the Logistics module. Previously, the *algresponsible* table hold information about responsible codes and the *resource_id* (not the *user_id*) that was linked to the code.

Agresso 5.5 introduced a solution where a responsible belongs to a certain role, and where role membership is based on the *user_id*.

Upgrade of orders created by Agresso 5.4x

The main purpose of this Responsibles Upgrade, is to convert active orders (created in the various logistics modules) in such a way that the responsible person still can be identified when the order is further processed.

Note

The upgrade will *not* convert the old responsible resources into new responsible *roles*.

Upgrade tasks

The main upgrade tasks are performed by the responsible upgrade wizard, and are described as follows:

Task	Description
Automatic code matching	Match resource_ids from the algresponsible table with the 5.6 users. If a user can be uniquely identified by a resource_id, the upgrade wizard will establish a new connection between the 5.4 responsible code and the 5.5 user id.

Manual matching	For all unmatched resource_ids, you are prompted to manually enter user_ids. There is no requirement that the new user really takes part in 5.5 responsible role. This task can be repeated several times, until all (old) resource_ids have a valid match.
Automatic order upgrade	The wizard will search through all relevant orders and convert all orders with a responsible codes where there has been a successful match. Log: The result will be written to the log, and can be viewed in the table <i>algprophead</i> .
Manual order update	For all the orders that were not converted, you will have to use the registration windows for the various order types, to set the correct responsible. Note: This requires that the various responsible roles are set up in the Responsible Setup window in the Smart client. See <i>Agresso 5.5 Release Notes, Logistics – Common logistics</i> .

Run the upgrade wizard

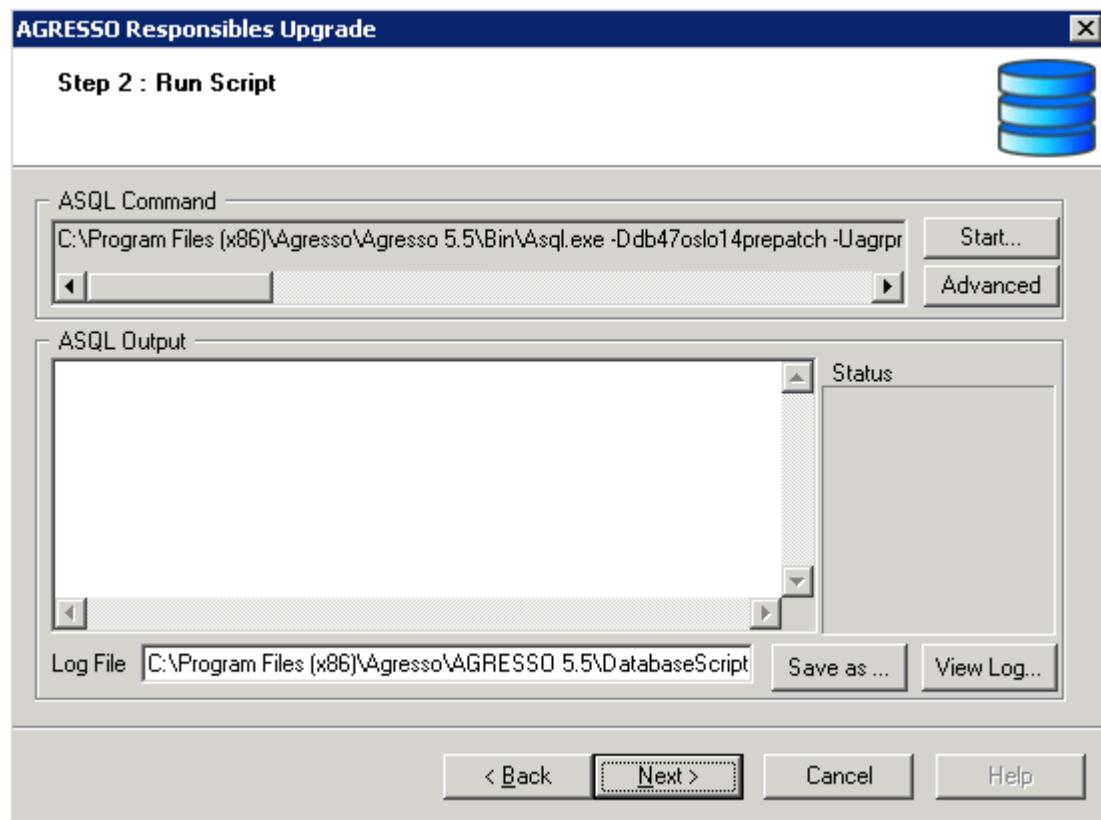
When the **Responsibles Upgrade wizard** is up and running, navigate to the Step 1 window, and do as follows:

1. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

You are now ready to run the upgrade scripts.

Note: If you already have run the upgrade scripts once, and in addition performed some manual matching (3. below), a re-run of the wizard will remove all manual updates!

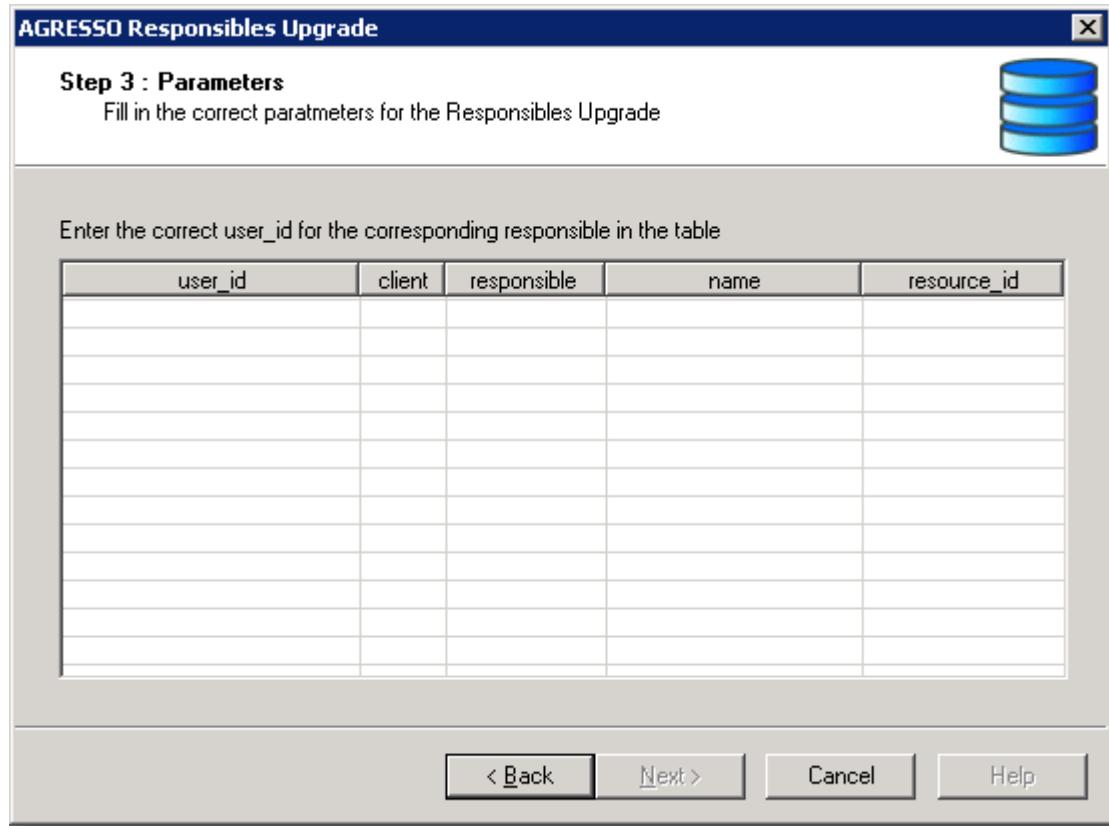
Responsibles Upgrade - Step 2



2. Click **Start** to run the script, and then **Next** (we assume that everything went OK after **Start**). The wizard will match all identifiable resource_ids with the 5.5 user_id. When completed, the status for the *algresponsible* table is set to S.

You will now be prompted to manually enter missing user_ids:

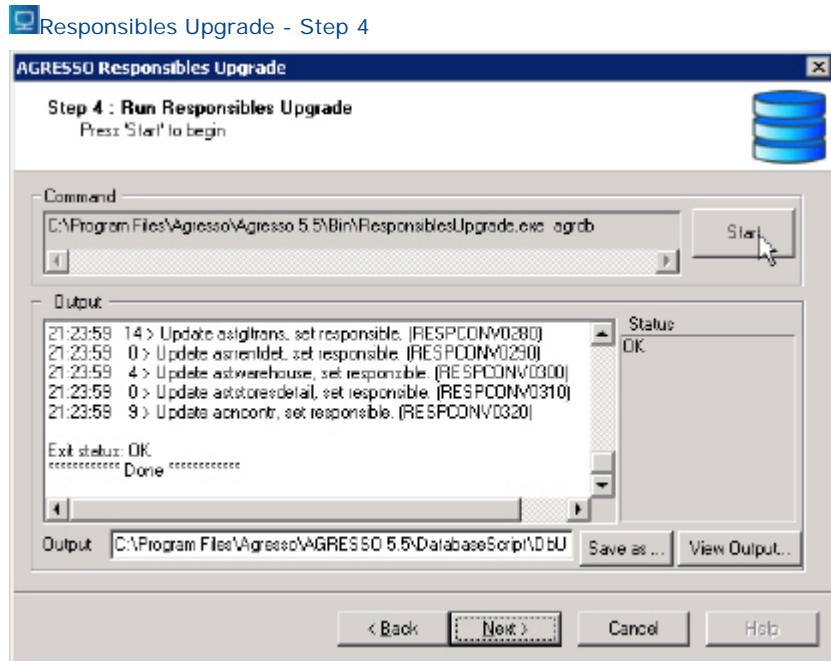
Responsibles Upgrade - Step 3



3. Repeat steps a. and b. as many times as necessary:

- Add missing user_ids.
- Click **Apply**.
- Click **Next** when you are finished.

The next task is automatic order upgrade.



4. Click **Start**, and then **Next** to finish.

This wizard can be run several times if you would like to add user_id's at a later stage.

Complete upgrade - additional manual tasks

There will normally remain some unmatched responsible codes on the orders. If so, the responsible roles must be registered in the Agresso Smart Client, and the orders must be updated in their respective registration windows.

BALANCE UPGRADE

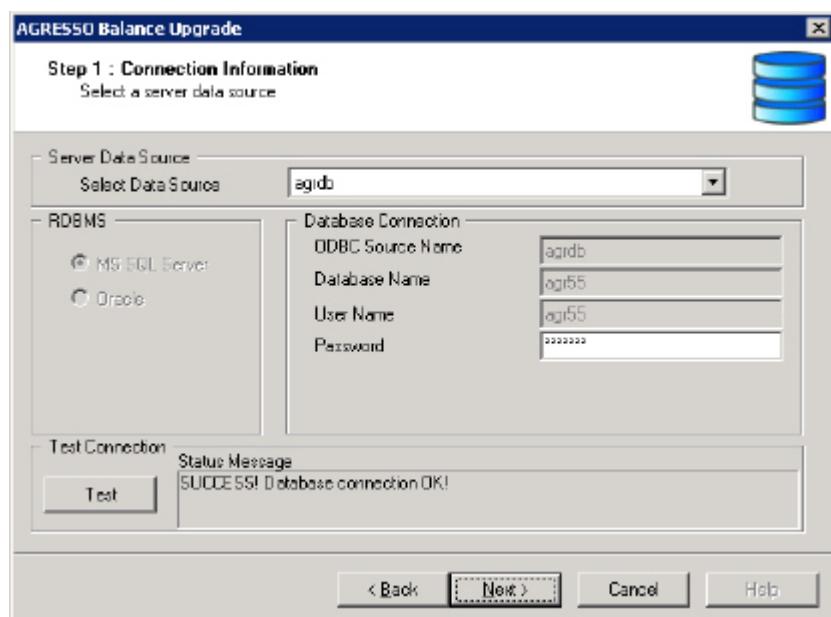
New solution for balance handling

Prior to Agresso5.5, the modules Project and Payroll had different solutions for balance handling. Agresso 5.5 (continued in 5.6) introduced a new, common solution, and old Balance data must therefore be converted.

Procedure

When the wizatrd is up and running, move to Step 1.

Balance Upgrade - Step 1

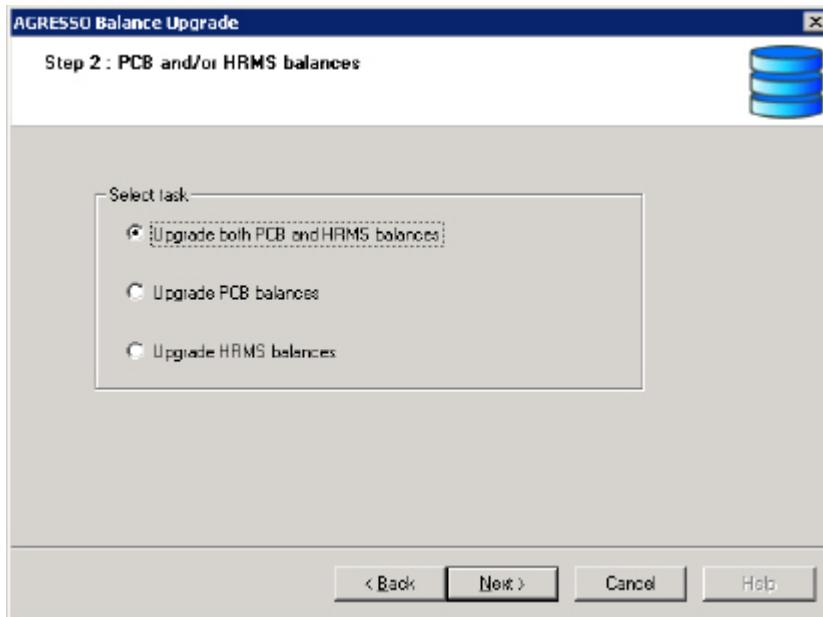


Do as follows:

1. Select correct data source from the drop-down list, fill in the correct password and click **Next**.

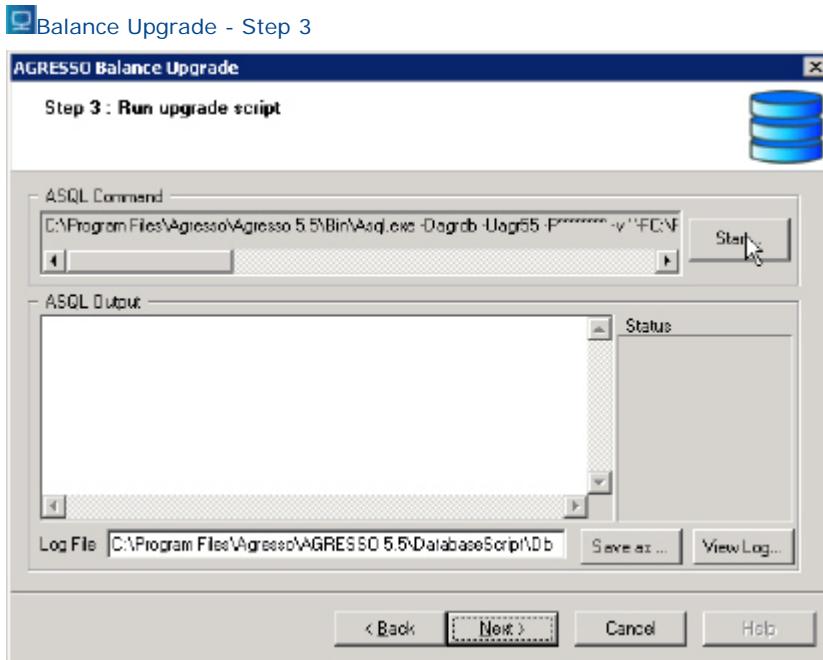
You will be prompted to select upgrade option:

Balance Upgrade - Step 2



2. Make your selection and click **Next**.

You are ready to run the selected upgrade script which will move date to the new *ahsbalance* table.



3. Click **Start** to run script, and then **Next** to continue.

The upgrade is completed and you can close the wizard.

WORK SCHEDULE AND ABSENCE DETAILS UPGRADE

Changes in 5.6

Previously both TR and HRMS had separate setups for work schedule. From version 5.5, these are merged into one common setup. Moreover, to use absence, the employees now need to be connected to a work schedule (through relation, system parameter or position). Previously this was optional.

The data model has changed. Absence is now stored per day in a new absence details table, which is not the case for already registered absences. To support reversing, reporting and enquiry purposes, the old absences must be converted.

Converting absence entries

To convert old absence entries, you need to decide whether you want to create work schedule entries for the employees' old overrides of personal work schedules (previously stored in apscalendar). Previously you could override the employees personal work schedules upon absence entry. If you need the history of this to be incorporated in the employees work schedules, you must specifically select to do that (see step 8 below).

Note: Beware that if you have many and old overrides, this will create many rows in the work schedule tables. It might be wise to clean out very old data before converting.

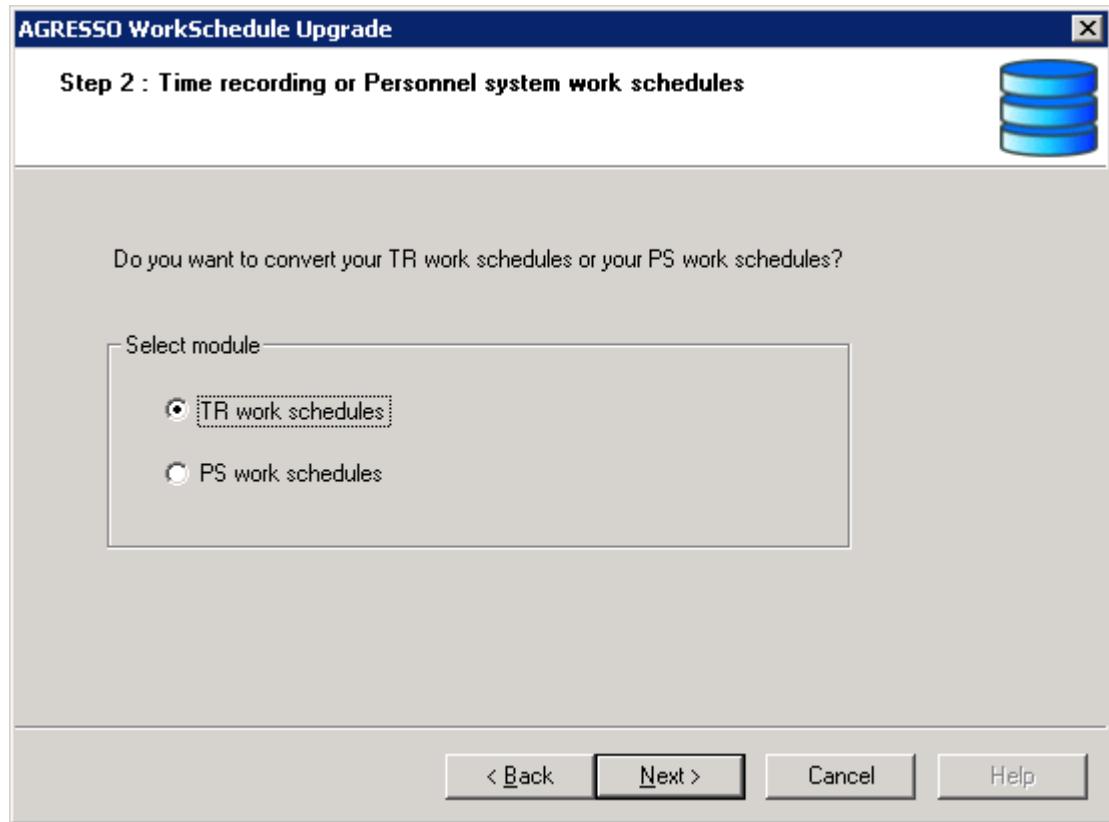
Upgrade

You use the Agresso Work Schedule Upgrade wizard on the Upgrade DVD to run the upgrade scripts. This is not a straightforward process, however, since the wizard requires that a system parameter is correctly set, and that the server processes **HS05** and **HS04** are run between two wizard steps.

1 Start the Agresso Work Schedule Upgrade and click **Next** to proceed to Step 1.

2. Enter connection information and click **Next**.

Work Schedule Upgrade - Step 2

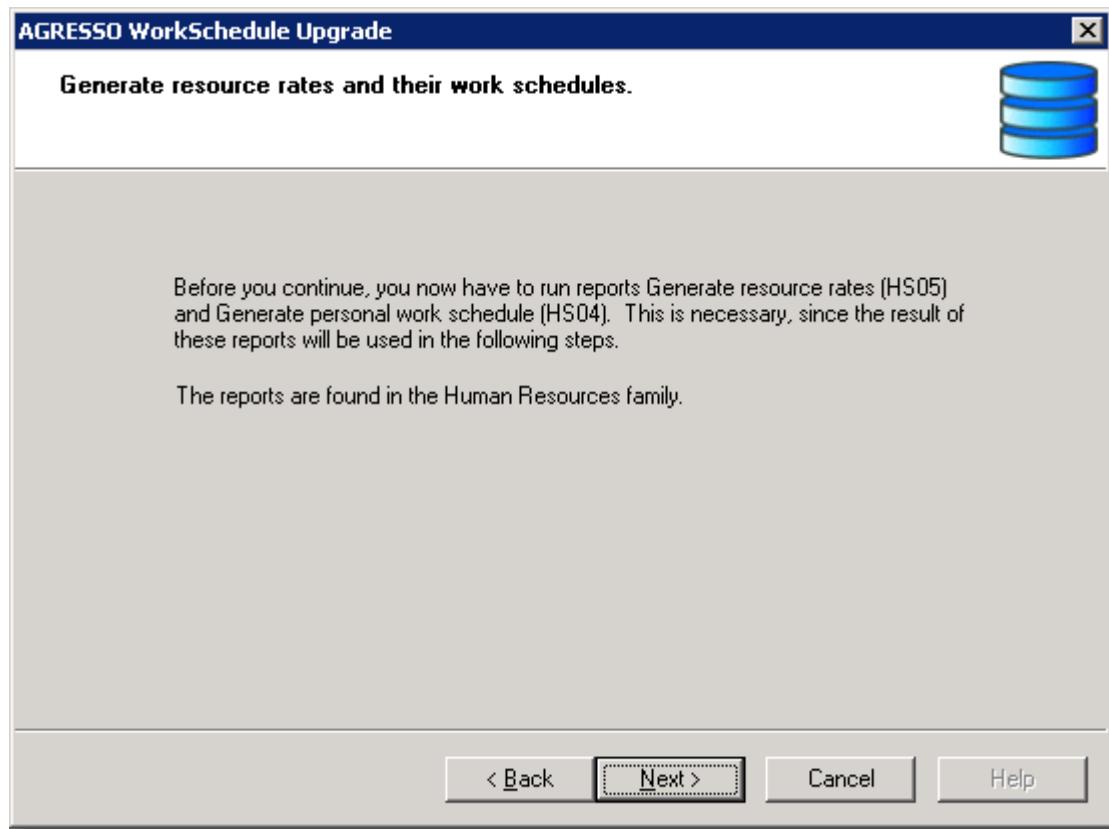


3. Select the schedules to upgrade and click **Next**.

You will now be ready to run the upgrade script for the selected schedules.

4. Click **Start** to run the script and then **Next** (if errors, see the log file in the ASQL Output window).

Example: Update ABW



You now need to complete the upgrade from the Agresso Smart client.

5. Start the Agresso Smart client and make sure that the system parameter HS_PARTTIME_PCT is set to the value reference you use for the employees part time percentage.

The screenshot shows a software window titled "System parameters - Payroll/Personnel/Expense". The main title bar says "EN System parameters - Payroll/Personnel/Expense". Below the title, there are tabs for "Setup", "Template", "Reset parameter to system setup value", "Reset parameter to default value", "Add Link", and "Organise Links". The "Setup" tab is selected. The main area is a table with columns: Sys.setup, EN, Client, EN, Name, Mod, Max length, Value, On/off, Sys. setup level, and Client level. The table contains several rows, including one for HS_PARTTIME_PCT which has a value of "25 I001".

Sys.setup	EN	Client	EN	Name	Mod	Max length	Value	On/off	Sys. setup level	Client level
?				2 ABS_DAYS_BETWEEN	HS	8		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	HS_CO_STATUS	HS	60 status	6 HS_DAY_BREAK_TIME	HS	4 0000	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7	HS_DEF_DAY_VAR	HS	2 N	8 HS_DEF_WORKSCH	HS	25	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	HS_MARK_DAY	HS	99 Sunday	18 HS_PARTTIME_PCT	HS	25	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

6. Run **HS05 Generate resource rates** with the following parameter settings:

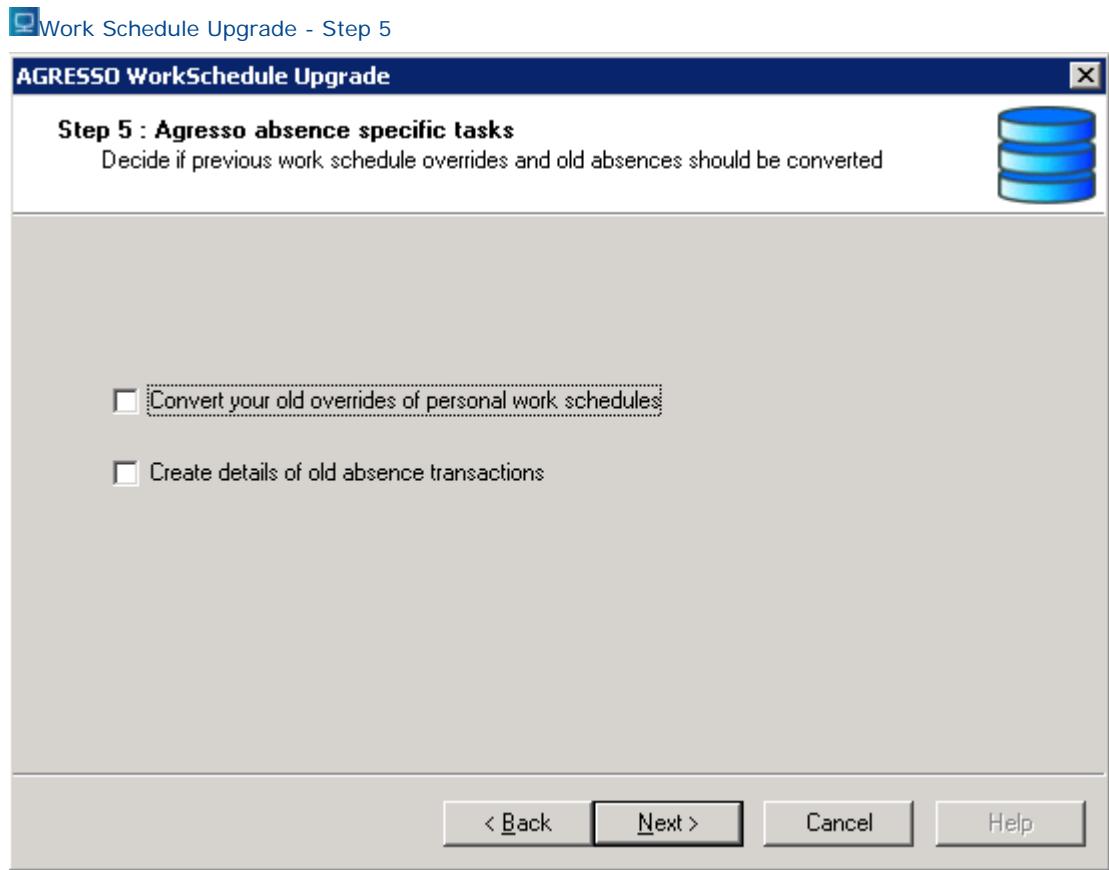
- Set the **Date from** as far back in time as you need to create rate history for the employees.
- Leave the **Relation** parameter blank and set **Relational value** to *****.

All rates will be converted to the new system of resource rates. This is essential to make Agresso HRMS run like before, regardless of the work schedule upgrade.

Note: Before you proceed with the next step, be aware of the following: If you want to use a holiday scheme to be included in the creation of the different work schedules, this must be set up in the **Holidays** window, before you run **HS04 Generate personal work schedule**.

7. Run **HS04 Generate personal work schedule** for all work schedules.

When step 7 is completed, you must return to the upgrade wizard and make a decision on absence specific tasks. See Converting absence entries above.



8. Make your selection and click **Next**.

You are ready to run the final scripts.

9. Click **Start** to run the scripts, and then **Next**. (If errors, see the log file in the **ASQL Output** window).

The upgrade is completed and you can close the wizard.

TIMESHEET UPGRADE

Introduction

The Timesheet upgrade wizard will upgrade the time transactions from both *group timesheets* and *personal timesheets*, based on the parameters you choose. The wizard is available on the Upgrade DVD. All time transactions will be stored in new, common tables, and before they are converted to the new data structure, you will not be able to view them in the timesheet windows.

Note: Time transactions that have *not* been updated can still be viewed using a browser enquiry against the 5.4 tables.

Parameter information

The wizard will upgrade timesheets for selected periods. You have to enter period from, posting period, and for the Group timesheets you will also enter last complete period and attribute id. The period from needs no further explanation, but the other three require a few words:

Posting period: The period will be used for selection voucher numbers for the timesheets. The voucher type used in the upgrade is the value of the system parameter VOUCHER_TYPE.

Last complete period: For this period and all previous periods, the timesheets status will be set to Terminated if all rows are terminated. If not, the timesheet status will be set to Draft. The rows which are not terminated will also be set to Draft. For timesheets after this period the status will be set to Draft for both the timesheet header and rows.

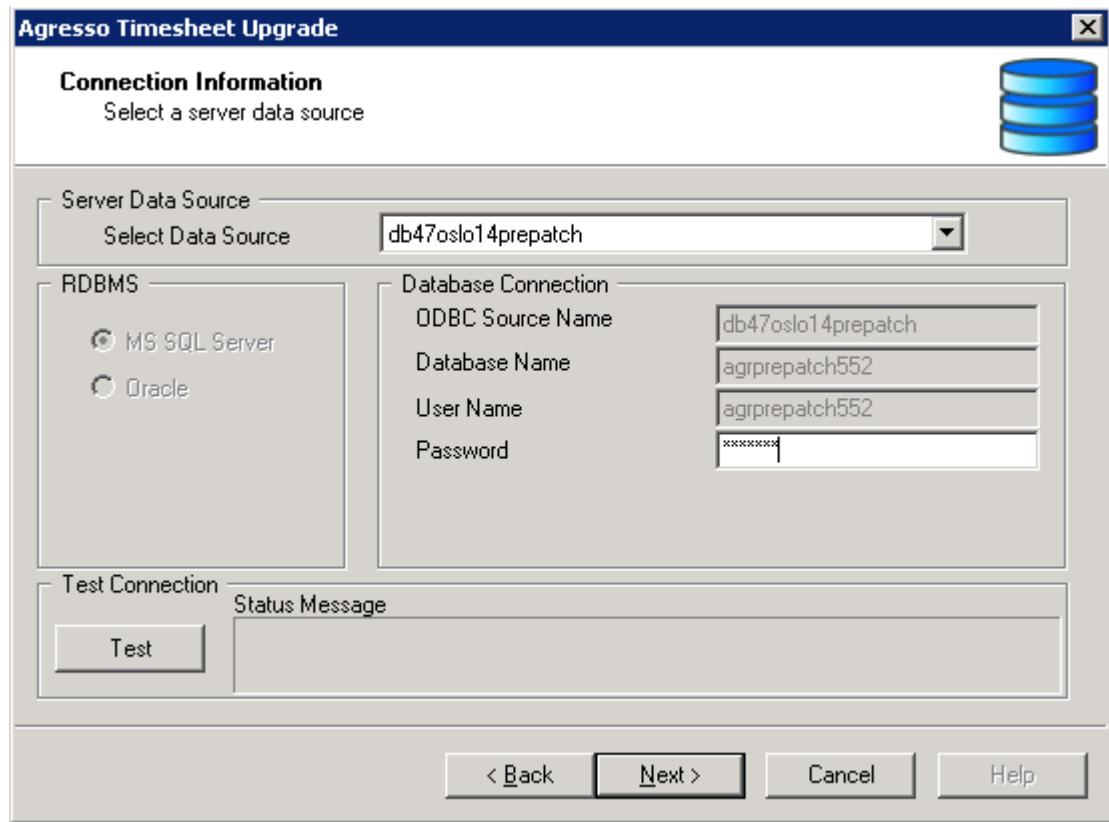
Run the Upgrade wizard

Before you continue, you must have the **Timesheet Upgrade wizard** up and running

Connect to database

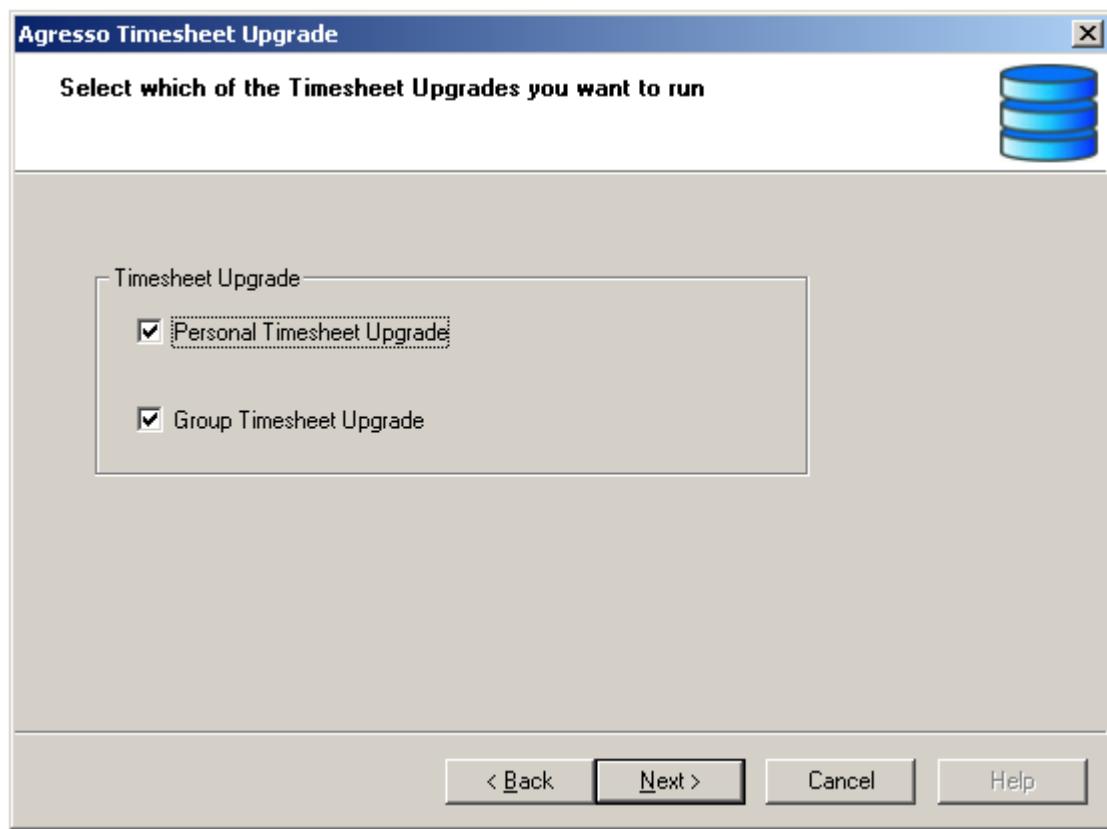
1. Navigate to the Step 1 window.

Example



2. Select the correct data source, enter correct password, and click **Next**.
You are ready to select what to upgrade.

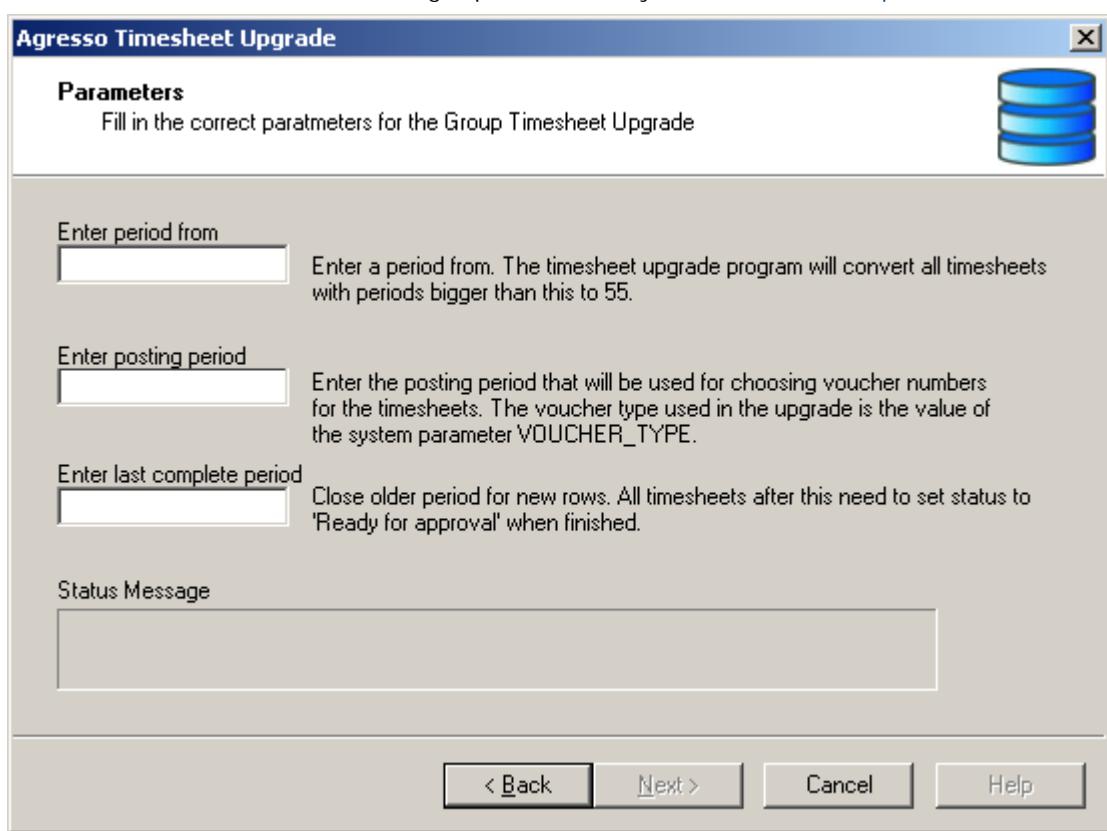
Example



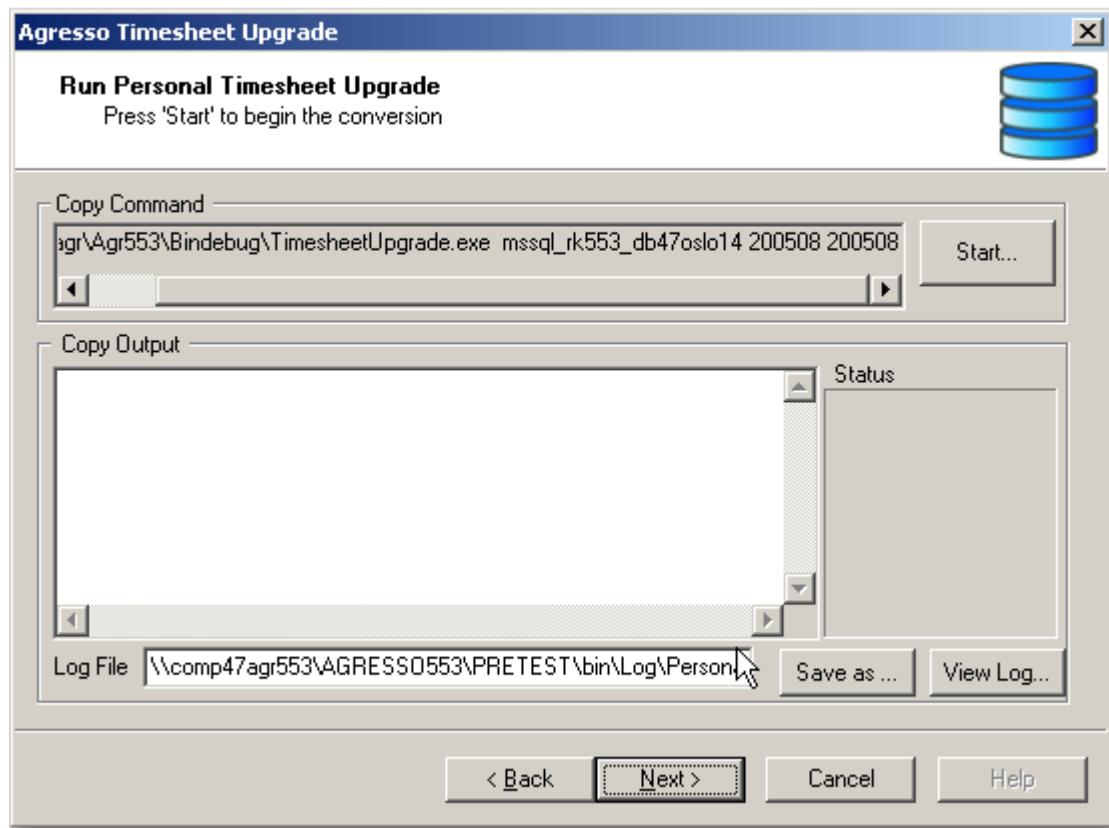
Select items to upgrade and run scripts

3. Make your selections and click **Next**. You are prompted to enter periods.

Note: The number of fields in the dialog depends on what you selected: [Example](#)



4. Fill in correct periods and click **Next**



5. Click **Start** to start upgrading.

6. Click **Finish** to complete the timesheet upgrade.

COPY SETUP BETWEEN DATABASES

Not to be used when upgrading to ABW 5.6!

Please note that the procedures described below is currently not valid! Send a request to the R&D forum.

General description

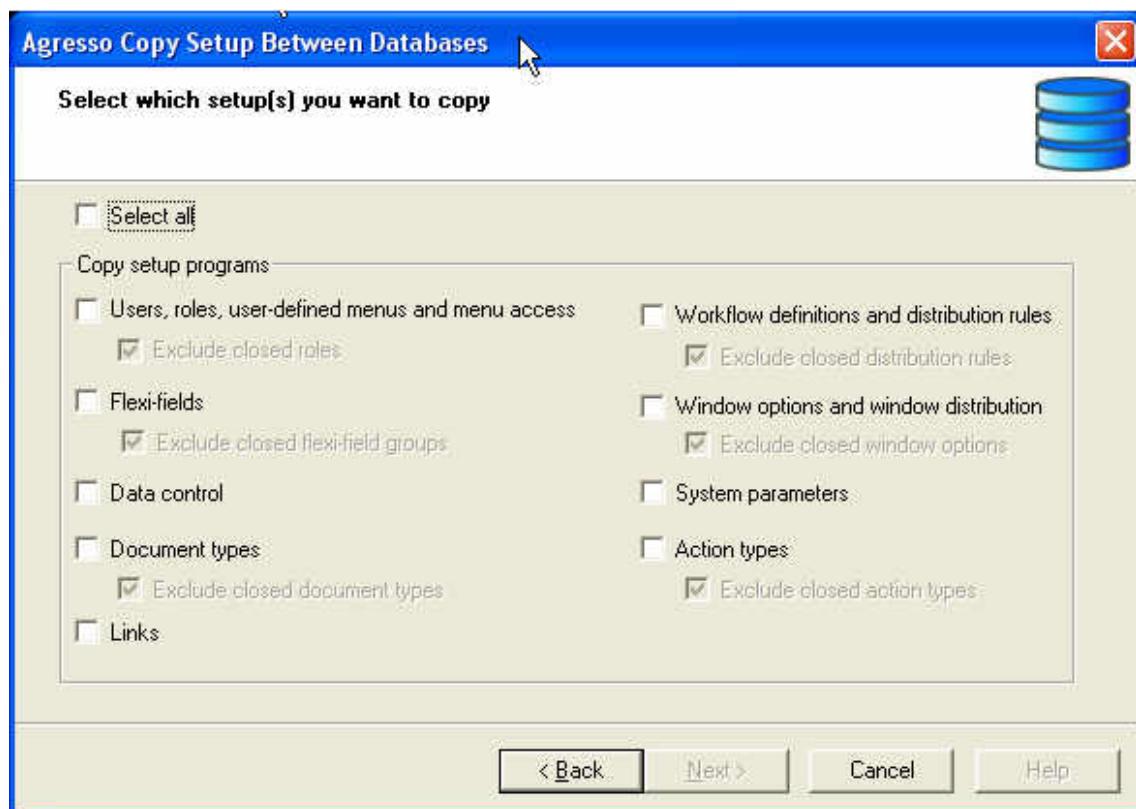
The **Copy Setup Between Databases** wizard copies the Agresso setup from one database to another. The following areas may be copied:

- Users, roles and menu access
- Workflow definitions and distribution rules
- Flexi-fields
- Window options and distribution of window options
- Data control
- System parameters
- Document types
- Action types

Note: Existing setup in the target database will be overwritten. See Dialog example below.

Dialog Example

The various options are presented in a single window:



Intended use

The wizard is created to facilitate transfer of a new and working configuration in the test environment, to the production environment.

Used in this context, it is important that all other updates are completed – on the test database – before you run the **Copy Setup Between Databases** wizard.

Run the wizard

Note: If you need more details before you run the wizard, see [Wizard details](#) and [Updated tables](#) below.

When the wizard is up and running, do as follows:

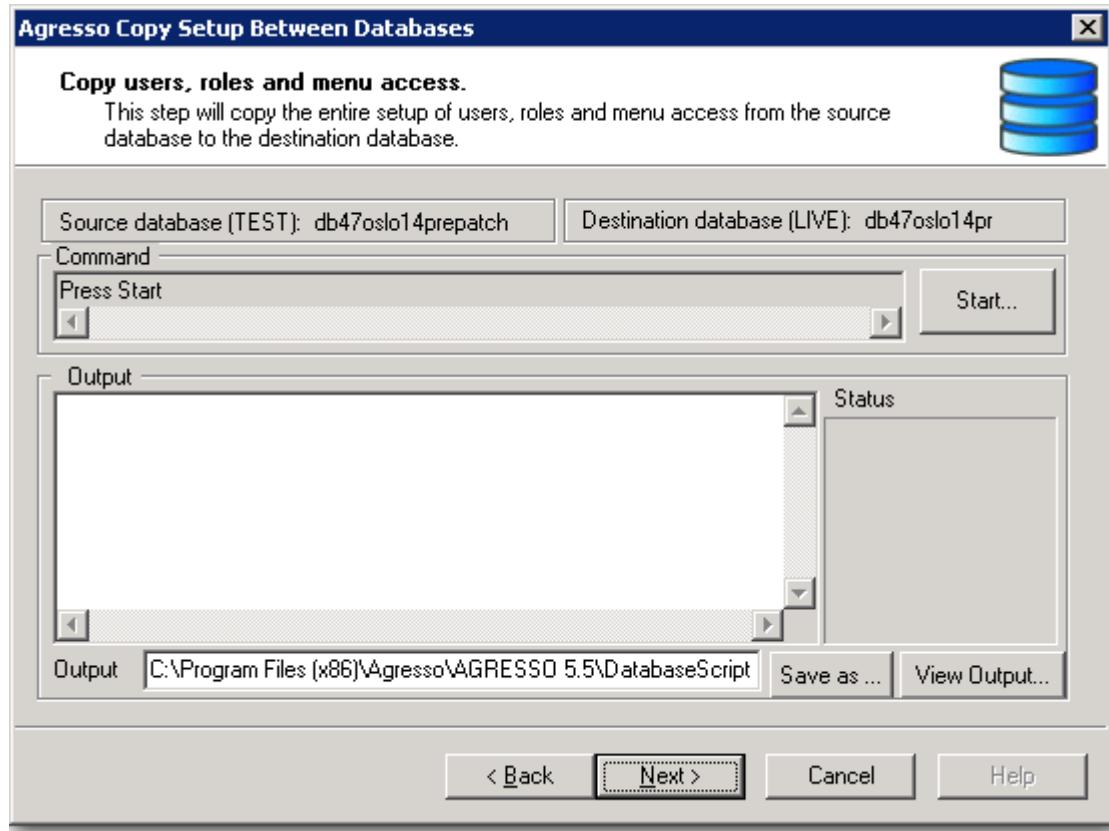
1. Enter connection information for the *Source* database and click **Next**.
2. Enter connection information for the *Destination* database and click **Next**.

You are presented with a few copy options. See Dialog example above.

3. Make your selections and click **Next**. The wizard is ready to start the Copy operation.



[Example.](#)



4. Click **Start**. You will need to confirm copying for each selected area.

Wizard details

The various copy programs (wizards) are described below:

The wizard ...	Will copy the following from source (test) database to target (production) database ...
Users, roles, user-defined menus and menu access	<ul style="list-style-type: none"> • New users (aaguser) with status Active • User information (addresses and passwords) connected to new users with status Active • All user detail information (aaguserdetail) • All roles • All menu access (aagaccess) granted to both users and roles • All user-defined menus (aagmenu) <p>You can exclude roles with status Closed from being copied to the destination database (default).</p>
Workflow definitions and distribution rules	<p>The entire setup of workflow and distribution rules. Existing setup will be replaced.</p> <p>You can exclude closed distribution (default).</p>
Flexi-fields	<p>The entire setup of Flexi-fields. Existing setup will be replaced.</p> <p>You can exclude closed Flexi-field groups (default).</p> <p>Note: No Flexi-field data will be copied!</p>
Window options and window distribution	<p>The setup of window options and distribution of window options. All existing setup will be replaced.</p> <p>You can exclude closed window options (default).</p>
Data control	<ul style="list-style-type: none"> • The entire setup of data control. All existing setup will be replaced.

	<ul style="list-style-type: none"> • All relations and relation values connected to attribute A11 –ROLEID • All setup made in Data control management (CR49). <p>Note: The program also updates all attributes with data control activated in the <i>source</i> database.</p>
System parameters	<p>System parameters. All changed parameters will be listed in a report file. The path to this report file is specified in the program's log file.</p>
Document types	<p>The setup of Document archive. All existing setup will be replaced. Note: No documents will be copied!</p>
Action types	<p>The setup of Action overview. All existing setup will be replaced.</p>
Links	<p>The setup of Links. All existing setup will be replaced.</p>

Updated tables

Below we list the tables affected by the copy programs, and indicate the operations they are exposed to:

Users, roles, user-defined menus and menu access

The following tables are affected:

Table name	Data replaced	Rows inserted	Rows deleted
awfelemttype	X		
awfelemtypedet	X		
awfelemtypelogrouping	X		
awfelemtypeitem	X		
awfelemtypemap	X		
awfprocaction	X		
awfprocdeadlines	X		
awfprocdelay	X		
awfprocdelaydet	X		
wawfprocelemtype	X		
awfprocess	X		
awfprocfunction	X		
awfprocrole	X		
awfprocsplit	X		
awfrule	X		
awfruledet	X		
awfrulegroup	X		
awfversion	X		
acrdiagramlayout	X		
awfalternate	X		
awfbmethods	X		

awfcolumns	X		
awfelemtypemenu	X		
awfmanstep	X		
awfprocrcd	X		
awfprocsin	X		
awfuserdetail	X		
aimblob		X	X

Flexi-fields

The following tables are affected:

Table name	Data replaced	Rows inserted	Rows deleted
acractionelemtype	X		
acractiotype	X		
agldimvalue		X	X
agldescription		X	X
acraktioncontact	X		
acraktionattval	X		
acracctionattvaldet	X		
awfelemtypede	X		
awfelemtypedet	X		
awfelemtypeitem	X		

Links

The following tables are affected:

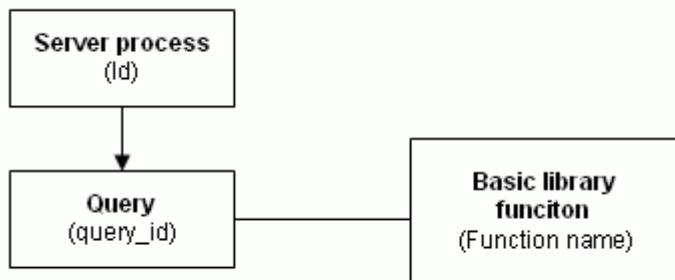
TUNING OVERVIEW

Running server processes

Queries and database operations

When you run a server process, a set of queries associated with the server process is executed. Each of the queries is identified by a unique query id (query_id), and each query will call a basic library function, identified by an Agresso function name.

This is illustrated in the diagram below:



The arrowhead illustrates that a server process executes one or more associated queries (the least complex process uses actually two queries) while the line between the query and basic database operation indicates that one query triggers one database operation.

The basic database operations

Agresso utilises six basic database operations, explained in the table below:

Agresso function name	Description
AGRCreateTableId	Creates a new database table. CreateTableId can also fill the new table with data defined by a select statement given as a parameter to the function (optional).
AGRExecSqlId	Executes any sql statement.
AGRCreateUpdateId	Creates a temporary table and uses it to do an update of another table.
AGRUpdTableId	Updates an existing table with data from (an)other table(s).
AGRSelIntId	Selects a record from one or more tables and returns the values into a program variable.
AGRSelLoopId	Executes a cursor loop given by a select (parameter). For each record of the cursor loop one or two statements are executed. These are also given as parameters. SelLoopId may typically be used to do an insert or update, depending on whether the record already exists or not.

Tuning options

A tuning option in our context is the possibility to force an extra sql statement to be executed in addition to the original (basic) query. The intention is to speed up processing. A basic query can have numerous tuning options (extra sql statements).

Example: You can insert an sql statement before the existing query is executed to create a new index and another to make sure that the index shall be used in the basic database operation. In addition you can insert an option after the original query has finished, to ensure that the tuned query is written to the log file.

Available tuning options

Reference: The descriptions in this section relates to options you have in the [Server Process Optimization window](#).

Agresso provides a set of tuning options, described in the table below:

Tuning option	Description
NO_OPT	Used to overrule any other tuning options.
LOGQUERY	Writes the query to the log file.

INDEX	Creates a new index for a given table, based on selected columns.
HINT	Allows you to add hints to the query optimiser. Most native hints from Oracle and Sqlserver are supported.
STAT	Updates available statistics tables.
PLAN (Oracle only!)	The query plan will be printed to the log file.
ABORT_JOB	Aborts the job before or after (dependent on value of Position) the query has run.
SNAPSHOT	Makes a copy of a temporary table created by the query. The table sequence number must be entered in the Table column. Default=0 (first)
STRATEGY	<p>Update strategy for query. Currently only relevant for Oracle 11.2 and later, when updating one table from another.</p> <p>You select the strategy in the Argument column:</p> <ul style="list-style-type: none"> • MERGE_ROWID - will use MERGE INTO with JOIN on ROWID • WHERE_EXISTS - will use JOIN in WHERE EXISTS • ROWID_IN - will JOIN on ROWID

Tuning option execution rule

The tuning option execution rule tells Aggresso when a specific tuning shall be executed in relation to the original query. There are three basic rules of execution, which will behave slightly differently depending on the basic database operation that is performed when the query is executed.

- | | | |
|---------------|---|--|
| Init | — | Option is to be executed as early as possible, before the actual execution of the query. |
| Before | — | For multi step operations, the option will be executed after the first step. |
| After | — | After the query is finished. |

Time of execution and database operation

The execution rules forces the following execution of the tuning option for each of the available basic database operations:

Database operation	Execution rule	Actual execution point
AGRCreateTableId	Init	Before table is created
	Before	Before data is inserted into the new table
	After	After table is created and data (if any) is inserted
AGRExecSqlId	Init	Before query is executed
	Before	Before query is executed
	After	After query is executed
AGRCreateUpdateId	Init	Before temporary table is created
	Before	Before update
	After	After update
AGRUpdateTableId	Init	Not relevant. Shall not be used.
	Before	Before update
	After	After update
AGRSellIntId	Init	Before query is executed
	Before	Before query is executed

	After	After query is executed
AGRSelLoopId	Init	Before the main select statement
	Before	Before the first query inside cursor is executed
	After	Before the second query inside cursor is executed
AGROpenCur	Init	Option is to be executed as early as possible, before the actual execution of the query.
	Before	For multi step operations, the option will be executed after the first step.
	After	After AGRCloseCur is called. I.e. LOGQUERY option will only log "Close Cursor", not the AGROpenCur query.

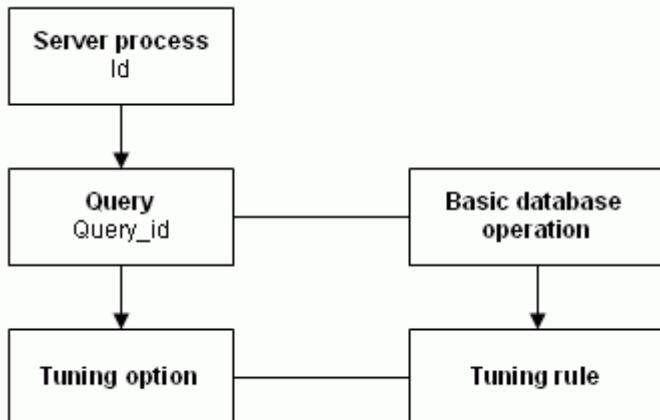
Tables used for tuning setup and modifications

The standard setup for the Agresso server processes is stored in the table asysqueryopt, delivered as part of the Agresso installation package. This table contains a limited number of tuning actions.

All tuning actions created for a specific customer are stored in the table aagqueryopt. Customer specific tuning will usually survive Agresso upgrades.

Tuning data model

The following diagram summarises the elements so far identified as part of the tuning process:



THE SERVER PROCESS OPTIMIZATION WINDOW

Window description

The Server process optimisation window is the primary tool to use for server process tuning. An example of the window is shown below, with default tuning for the LG04 server process, variant 0:

Window example

An example of the Server process optimization window is shown below:

(AGRESSO DEMO) Server process optimisation									
Report	LG04	Variant							
	Default	Query	SeqNo	Position	Description	Options	Argument	Table	Database
?	<input type="checkbox"/>								
1	<input checked="" type="checkbox"/>	LG04_HV0010	0	After		INDEX	order_id	0	All databases
2	<input checked="" type="checkbox"/>	LG04_HV0010	1	After		STAT	compute statistics	0	Oracle
3	<input checked="" type="checkbox"/>	LG040030	0	After		INDEX	order_id	0	All databases
4	<input checked="" type="checkbox"/>	LG040030	1	After		INDEX	apar_id,order_id	0	All databases
5	<input checked="" type="checkbox"/>	LG040030	2	After		STAT	compute statistics	0	Oracle
6	<input checked="" type="checkbox"/>	LG040040	0	After		INDEX	article_id,order_id,client	0	All databases
7	<input checked="" type="checkbox"/>	LG040040	1	After		INDEX	order_id, line_no	0	All databases
8	<input checked="" type="checkbox"/>	LG040040	2	After		INDEX	apar_id,order_id	0	All databases
9	<input checked="" type="checkbox"/>	LG040040	3	After		STAT	compute statistics	0	Oracle
10	<input checked="" type="checkbox"/>	LG040070	0	After		INDEX	order_id	0	All databases
11	<input checked="" type="checkbox"/>	LG040070	1	After		STAT	compute statistics	0	Oracle

Explanation of fields

Top row: The top row contains drop-down lists for the server process (labelled Report) and variant number you want to optimise.

Table columns: The table columns contain tuning details for the various queries. These are explained in the following table:

Column name	Description
Default	A flag indicating whether this row is part of default tuning or not.
Query	The query id identifying a specific query associated with the server process.
SeqNo	The sequence number of the tuning option. When a specific query id is repeated in several rows, the options are executed in the following order: 1. All options with Init as execution rule (in the Position column), then ordered by their sequence numbers. 2. All options with Before in the position column, ordered by sequence numbers. 3. All options with After in the position column, ordered by their sequence numbers.
Position	The execution rule to apply. See previous pages. Note that rules are implemented according to the basic database operation the query performs.
Description	A description of the tuning option.
Options	The selected tuning option. See previous pages.
Argument	The argument needed for the selected option. Descriptions are given below.
Table	A number identifying the table in the query.
Database	The selected database to which the tuning entry applies.

Appendix

TABLES USED BY AMC

The following tables are used by AMC:

aagserverqueue

Holds the configured server queues.

aagserviceuri

Contains the published web applications.

asysservice

Contains the available web services. The services appear in the "Add Web Service" wizard.

aagservcmdhead and **aagservcmddetail**

Contains commands posted to AgrBusinessServer service. For instance when a server queue is added from AMC, the change is added to this table, the AgrBusinessServer service polls these tables for changes, and executes the command, without having to restart the service.

aagsystemconfig

Contains Mail, SMS and printer configuration.

agouserconfig

Contains mail configuration for Agresso users.

aagalerthead and **aagalertdetail**

Contains status of jobs used by Process View in AMC and the Alert Server Web Service. The table: *acrrepord*, is used for monitoring the report queue in the Process View. By default these tables are polled every third second when the process view is open. The poll frequency can be configured from the properties dialog on the "Agresso Business World node" in AMC.

aagitwsservice, **aagitwsversion**, **aagitwsextension**

Tables that contains web services configured on the Agresso Web Service Host.

acrupgradehead, **acrupgradedetail**, **acrupgradeblob**

Used by the Upgrade Manager.

acrmalqueueeserv

Contains SMTP profiles configured for AgrMailQueue and the profile settings for the queue.

A NOTE ON CRYSTAL REPORTS

Crystal Report Runtime files

Crystal Report files are *no longer* part of the Agresso Business World installation.

If you have reports made by Crystal Reports, you must either copy the runtime files from an older ABW installation cd (version 5.4.2 or earlier), or contact your local support centre (Agresso Subsidiary or an Agresso partner).

The Agresso Logging System

The Logging Model

Introduction: Basic concepts

Destinations and formatters – an introduction

Configurable logging areas

Overview

Configurable elements

Destinations and formatters

Standard destinations

Formatters

The Agresso Logging System

The Logging Model

Introduction: Basic concepts

Log entries

A *log entry* is an ABW system term and the basis in ABW's logging system.

In Agresso Business World, all (unexpected) incidents as well as all database operations, generates a so-called log entry in the system. We can also use the term "loggable system incident" to refer to anything that generates a log entry.

So, when the ABW system detects a loggable incident, it will collect information about the incident and where it happened, and produce a log entry. Since all database operations, as well as several other "normal" operations in the system, are defined as loggable incidents, a huge amount of log entries are produced – all the time. If you wanted to write everything to a log (for example to a log file), it would clearly affect the processing capacity of the system.

This kind of detailed logging is not required during normal production, but it may be very useful if an unexpected error suddenly occurs.

So, everything that is written to a log, ultimately comes from *log entries*. But what kind of loggable incidents you really want to log, and when and how you want to log them, is a matter of configuration.

Two basic configuration properties

There are two basic log entry properties that are essential when you configure logging for the ABW system:

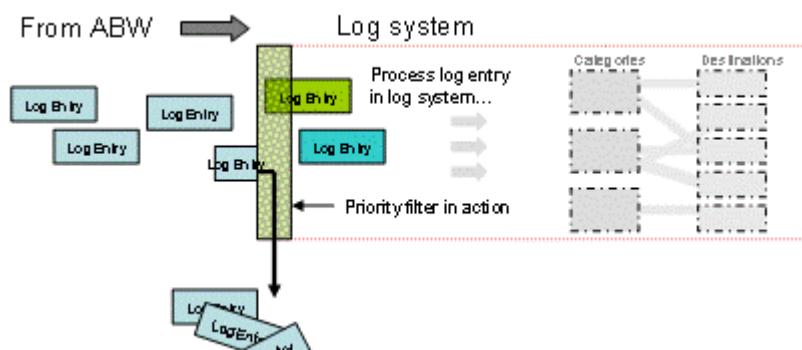
- the *priority* given to the log entry when it is created. The priority comes from the system and can not be affected by anybody. Currently, there are five different priorities in use.
- the *severity* assigned to the log entry when it is created. We have six different severity levels, corresponding to how seriously the loggable incident affects the system.

The *configurable part* of the logging system is to a large extent built around log entries, and their priority and severity.

Main log filter

When a loggable incident occurs in ABW, a log entry is created and sent to the logging system for processing.

When a log entry enters the logging system, it is first met by a so-called *priority filter*. The priority filter simply checks the priority of the log entry, and decides whether to let it through or not. If the priority set for the priority filter is higher than the priority set on the log entry, the log entry is immediately discarded. Accepted log entries are processed further:



Log categories

The basic properties belongs to the log entry, while the concept of *log categories* is part of the logging system.

A log category is set up to accept log entries with a certain *severity level*, meaning that all log entries with a severity level equal to, or higher than, the value accepted by the category, will be accepted and processed further. Log entries not accepted by any category will be discarded.

When a log entry is accepted, it will be written to a medium of some kind (for example a file), represented in the logging system as the log entry's *destination*.

ABW comes with three predefined log categories, intended to cover the most common logging needs:

- *Normal* – for "normal" logging. Will log errors and critical incidents (relatively "high" severity levels).
- *Detailed* – for more details. May be useful for special modules or server processes, or when you need to investigate something in more detail (already shown in the "normal" log).
- *Custom* – for advanced logging options.

Destinations for the log

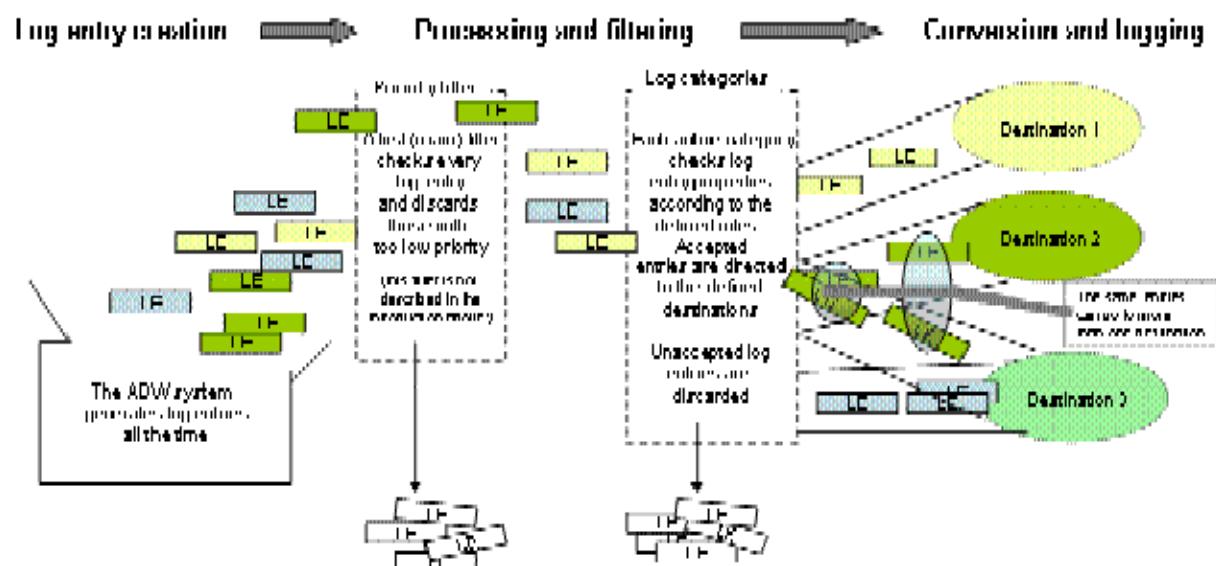
A log category will receive log entries all the time, but all log entries with incompatible property values will be discarded. The rest will be processed further, and written to one or more log destinations.

Log destinations: A log category will always be set up with at least one *log destination*, but it can be several.

A log destination defines the place to actually write the log content (to a file, to the database, to a message queue), and also *what to write* (the log entry holds a lot of information) and *how to format it*. It is all configurable.

From log entry to log – an overview

The diagram below illustrates the basic flow of log entries in the ABW system.



Destinations and formatters – an introduction

Log entry destination

A destination is the final end point for a log entry in the logging system (although it may not be the end-point for the log entry: an email destination will for example write the log entry to an external (mail) system, and the "final destination" is probably the mail recipient's inbox).

Technically, a destination is a component (a plug-in) capable of receiving and interpret a log entry, and write selected information to a defined medium.

Normally, this medium is a text file, but it can also be something else, for example a database table.

The logging system comes with a series of standard destinations, described later in this document.

Destination types

As long as there can be different destinations for a log entry (a text file, a database table, an email system, or Windows event log – to mention a few), there must also be functionality – tailored for the specific destination type – that converts the content of the log entry to a format which can be accepted by the destination. Unless this is done properly, the destination will not be able to accept the log entry.

Some of the standard destinations have a fixed (hard-coded) conversion mechanism, but most of them are configurable. In case of the latter, a so-called *formatter* component takes care of the conversion. A formatter also allows you to select the information you want to be written to the log and how it shall "look" (i.e. be "formatted").

Formatters

A *formatter* is a specialised component (implemented as a plug-in) which takes any log entry and converts it to a format suitable for a given destination type.

Standard ABW formatters are listed later in this document.

Configurable logging areas

Overview

Logging configuration page

In the **Agresso Management Console (AMC)**, all areas – or ABW applications – open for logging configuration, will have a separate page called **Logging Configuration** (behind the **Logging** tab when you open the associated node in **AMC**).

The **Logging Configuration** page is the same for all configurable areas, but will display headings, sections, and fields according to the current configuration of that area.

Logging areas (or logging applications)

The following areas (or applications, or programs or processes...) have configurable logging options:

- Agresso Business Server
- Server queues
- Web services, including the Self service client
- Smart client
- Central Configuration Server
- Applications based on Agresso Platform, added to Agresso Business World through AMC

A note about server queues (for those who are specially interested...)

The general logging configuration for server queues is stored in an xml file called **logger.config**, available in the **\bin** folder for the ABW installation.

When you add (or activate) a server queue in your ABW installation, and starts configuring the logging options for that server queue, your custom logging configuration for that specific server queue will be stored in a new file, called **<server queue name>.logger.config**. Subsequent modifications of the configuration will be written to this file.

Specific logging for reports handled by the server queue: If required, you can also edit your own **<server queue name>.logger.config** for a given server report, for example **GL12**, which is handled by the **DEFAULT** server queue. Assuming you already have set logging options for **DEFAULT**, the file **DEFAULT.logger.config** will store your settings.

You can now edit `DEFAULT.logger.config` directly, and modify it to suit your logging needs for `GL12`. When you save it as `GL12.DEFAULT.logger.config` you will create a custom logging configuration for all `GL12` reports.

Modifications by ABW users: When an ABW user now orders a `GL12` report, the default logging options will be taken from `GL12.DEFAULT.logger.config`. If this file doesn't exist, the default logging options will be taken from `DEFAULT.logger.config` – and so on.

If the user wants to modify the logging options for his or her order number, ABW will create a temporary file, called `<order number>.GL12.DEFAULT.logger.config`, used only one time.

Configurable elements

General tasks

When you configure an area for logging, assuming that you do not turn logging off completely, you will need to consider the following:

1. the minimum *priority* that shall be logged (priority filter value will block all unacceptable log entries)
2. the *categories* to use (selected categories will then be "active categories")
 - a. the severity level(s) for the active categories (log entries with "lowest acceptable" severity level – or higher – will be logged).
 - b. the *destinations* to use for the active categories (one log entry can go to several destinations)
 - c. the output and how it shall be formatted.

Priority filter

A log entry can come with any of the following values set for the *priority* property:

- Low
- BelowNormal
- Normal
- AboveNormal
- High

Setting priority filter: The priority filter will block all log entries with a value lower than the value you select.

Categories

There are three main *categories* for logging:

- Normal – to be used when we need to log errors, warnings and information events.
- Detailed – All operations are logged.
- Custom – to be used when the ABW user (administrator) knows exactly what to log.

Setting severity level for category: The severity level set for a category ensures that only log entries with the selected level – or higher – is written to the destination.

Destinations and formatters

Destinations and formatters require more detailed descriptions. See below.

Destinations and formatters

Standard destinations

Overview

The following destinations are delivered with ABW:

Name	Description
Flat File	Writes to a text file. When used for a specific log area, you can select the file name (location). and the encoding to be used. You decide the file size and how many backlog versions you want to keep, before old files are replaced with new ones. In other words, you can configure the "rotate" rules.
Detailed Flat File	Another text file destination, with one additional option: add extended logging information.
Custom EventLog	For "custom" logging to the user defined eventlog. Will by default log detailed log entries in "trace" style. Both eventlog destinations (this and the one below) have a separate minimum priority filter. The minimum priority for events is by default set to High.
Health Monitoring EventLog	For Standard event logging of documented events, recognisable by their Ids. The events are logged to the windows "Application" event log. It also has a special filter for excluding certain Ids. 3rd party tools can be used to monitor the eventlog and take actions based on the information written to the eventlog.
Server	In principle the same as Flat File, but with a fixed location (file) in the AGRESSO_LOG directory with filename: <code><reportName><orderNumber>.log</code> . It is used as default by the <i>Report</i> queue, and should not be used by other log areas.
Syslog	The Syslog destination adds support for writing to a syslog destination. (http://en.wikipedia.org/wiki/Syslog). The Syslog server should be configured to listen to the UDP port configured in the Agresso syslog destination.
Debug	A variant of Detailed Flat File. The Debug destination is the synonym for the standard application output destination. For example: If running a commandline application, debug destination should be used if you want to write log messages to the console (screen). Debug is normally not used in a production environment.
MessageQueue	Writes to Microsoft Message Queue. You must set the log path, and select the log entry properties you want to be part of the log. You can view the log entries via Computer Management . Note: By using the DbLogService server queue (under Server Queues in AMC), you can move the Log entries directly into a database table ((acrlog)), and view them via Agresso's Database log analyzer (comes with the installation). See More about Message Queues below.
ProcessMonitor	Writes to Microsoft Process Monitor. See below .

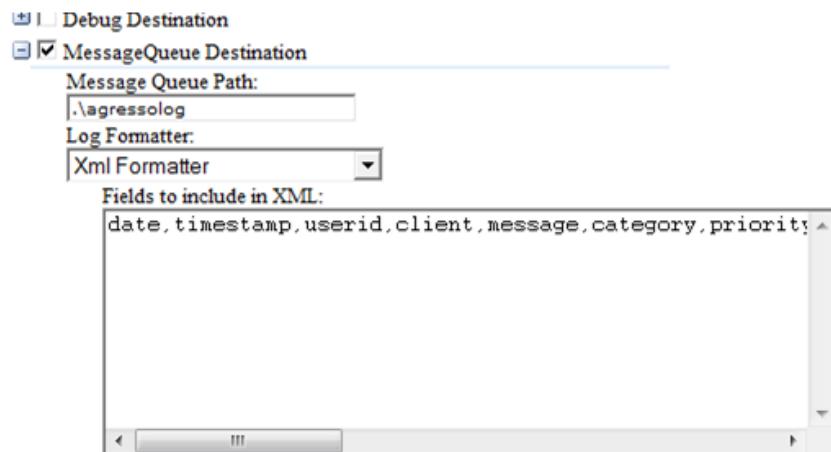
More about Message Queues

When Microsoft Message Queue is installed on the server (see <http://msdn.microsoft.com/en-us/library/aa967729.aspx>), you can also utilise it for efficient logging.

Basic settings: When you select MessageQueue Destination, you need to

- specify the name of log path.
.\\ gives you the default location for the current machine (Services and Applications / Message Queuing / Public Queues),
- select XML formatter as Log Formatter,
- enter the fields to include in the log entry.

An example is shown below. Here we have entered the log path agressolog, which will be created on the current server.



All log entries will be available in Computer Management, under the new path:

Label	Priority	Class	Size	Message ID
Event 2037	3	Normal	677	7029cb39-d24e-45b8-99f8-57f97cbf5267 692
Event 2051	3	Normal	718	7029cb39-d24e-45b8-99f8-57f97cbf5267 693
Event 2051	3	Normal	697	7029cb39-d24e-45b8-99f8-57f97cbf5267 694
Event 2029	3	Normal	602	7029cb39-d24e-45b8-99f8-57f97cbf5267 695
Event 2030	3	Normal	660	7029cb39-d24e-45b8-99f8-57f97cbf5267 696
Event 2037	3	Normal	677	7029cb39-d24e-45b8-99f8-57f97cbf5267 697
Event 2051	3	Normal	697	7029cb39-d24e-45b8-99f8-57f97cbf5267 698
Event 2037	3	Normal	683	7029cb39-d24e-45b8-99f8-57f97cbf5267 699
Event 2051	3	Normal	700	7029cb39-d24e-45b8-99f8-57f97cbf5267 700
Event 2037	3	Normal	683	7029cb39-d24e-45b8-99f8-57f97cbf5267 701

Log to another machine: If you want to send the log entries to a message queue on another machine, you need to replace the `.\` with the fully qualified machine name, e.g.
`theotherserver.corp.orate.com\agressolog`.

Move log entries from message queue to database: You can use **AMC's DbLogService** to move the log entries from the message queue and into the database (table `acrllog`).

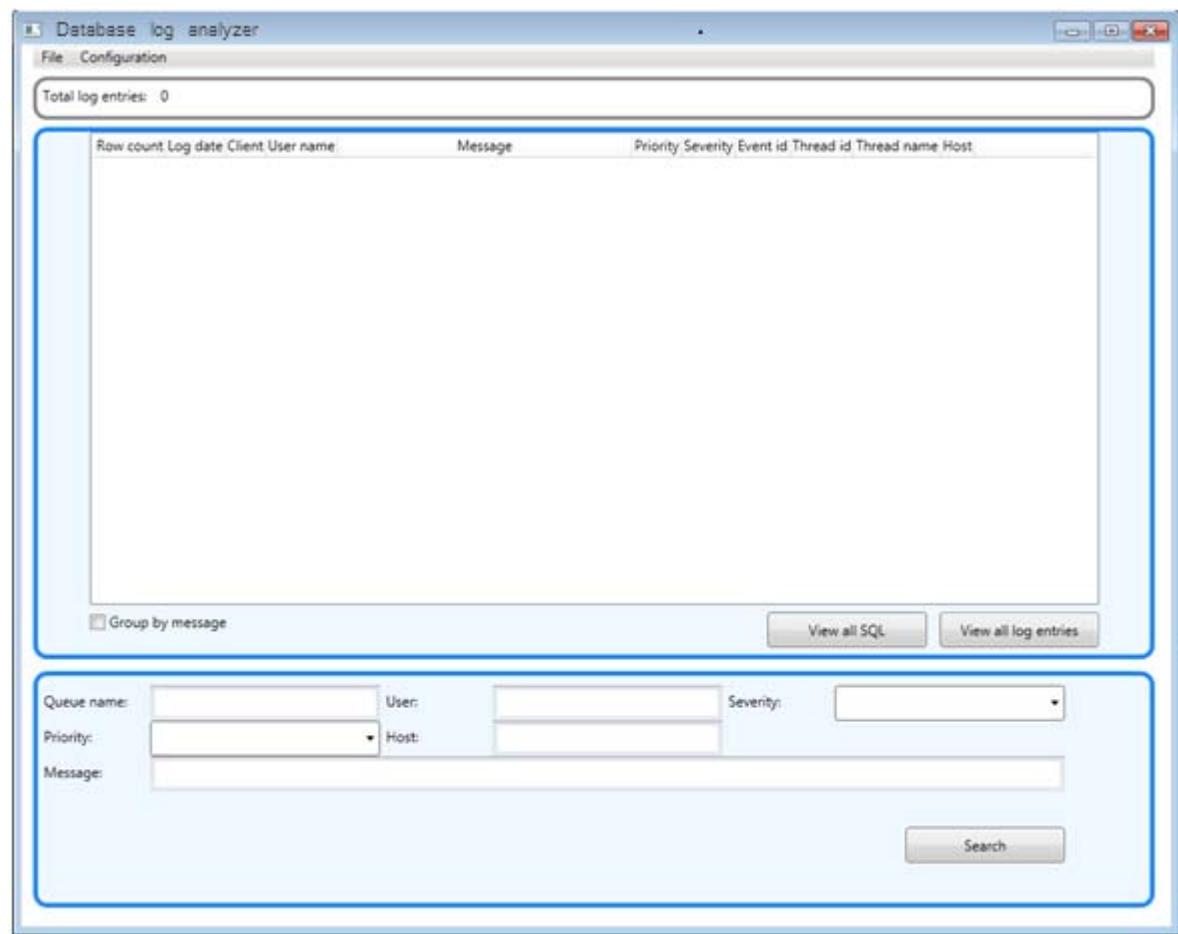
DbLogService, when activated, will check the defined message queue path regularly, and move the events into the `acrllog` table, as soon as they appear.

Note: The `Parameters` field must contain the exact same path as set up for the MessageQueue Destination, prefixed with `-m`.

So, if the `Message Queue Path` is `.\agressolog`, the `Parameters` value for DbLogService must be `-m.\agressolog`.

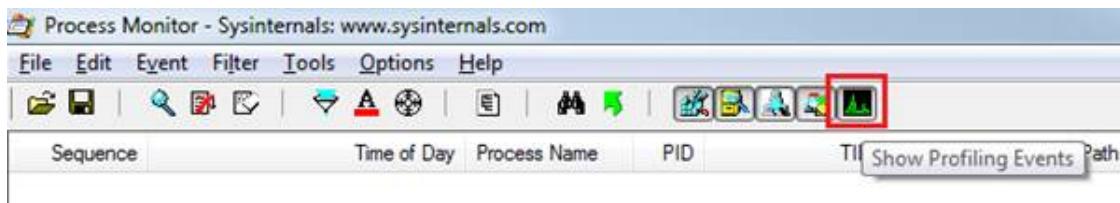
Consequently, if the `Message Queue Path` is `theotherserver.corp.orate.com\agressolog`, the `Parameters` value for DbLogService must be `-mtheotherserver.corp.orate.com\agressolog`.

Agresso comes with the Database Log Analyser tool, tailored to view log entries in the database. You find it in the server's `\bin` directory, named [DatabaseLogAnalyzer.exe](#):



More about Process Monitor

Microsoft Process Monitor is an advanced monitoring tool that shows real-time file system, Registry and process/thread activity. Agresso log events can also be shown in Process Monitor by enabling Show Profiling Events (not enabled by default)



Agresso log events will show up as **Debug Output Profiling** in the *Operation* column.

Process Name	PID	TID	Operation	Detail
Agresso.exe	13144	9048	ReadFile	Offset: 1 024, Length: 17 920, I/O Flags: Non-cached
Agresso.exe	13144	9048	Debug Output Profiling	Output: 14:28:55 Starting Agresso Smart Client
Agresso.exe	13144	9048	CreateFile	Desired Access: Read EA, Write EA, Read Attrib

For more details, see: <http://technet.microsoft.com/en-us/sysinternals/bb896645.aspx>

Formatters

Available formatters

A formatter can be associated with a destination of a certain type, compatible with the destination medium. If the formatter is available for configuration, you can define the content and layout of the final log.

Currently, Agresso provides three, configurable formatters:

- TextFormatter – for pure text formatting. This is used for most output of human readable text and it is used by several destinations.
- Detailed Text formatter – a variant of TextFormatter. Has some predefined settings.
- Xml Formatter – for the MessageQueue destination. You can select the properties you want from the log entry to be written as xml nodes to the log.

Tokens for text

The two formatters for text are both based on the same formatter component (Agresso.Plugin.Formatter.TextFormatter).

Both allows you to select the properties from the log entry that you want to be written to the log, by adding or removing so-called *tokens* to the configuration file.

Each log entry property is represented by a token, which is a string value on the form {<property name>}.

The following tokens (and properties) are available:

Token	Description
{date}	Current date (yyyy-MM-dd)
{timestamp}	Current time (HH:MM:ss)
{message}	The actual log message
{category}	The category name.
{priority}	Priority of the log message
{eventid}	Event id. ID for the type of error (required when using a destination that writes to the event log)
{severity}	The severity level
{title}	The title of the log message
{errormessages}	Any error messages

{machine}	The name of machine the application is running on
{appdomain}	Name of application domain
{threadname}	Thread name (program that caused the log entry to be created)
{threadid}	Thread Id
{activityid}	Activity id
{processid}	Process id
{processname}	Process name
{extra}	Some additional information added by system when the log entry was generated.
{parameters}	The parameters of a launching application
{userid}	Agresso user id
{client}	Agresso client
{eventcategory}	The Windows event category (only to be used for the Health Monitoring EventLog destination).

The TextFormatter's template property

TextFormatter uses a template to format the information. The template is simply a template string, consisting of the tokens and any other (fixed) text you may want to put into the log file. If you don't provide a template for the formatter in the configuration file, it will use a default one.

AMC provides you with a separate page for template configuration.

Node for the Xml formatter

The available Xml nodes corresponds to the tokens, but are written without the parenthesis ('{', '}').

CONDITIONAL FUNCTIONS: CASE

CASE

CASE returns a value which can be used in ASQL statements.

Syntax

The syntax is as follows:

```
CASE [ expression ]
    WHEN condition_1 THEN value_1
    WHEN condition_2 THEN value_2
    ...
    WHEN condition_n THEN value_n
    ELSE value_default
END
```

Example

The following example shows how you can use CASE to simplify your SQL. The original statements were as follows:

```
UPDATE acrtrans SET status = 'N' WHERE status= 'Y' AND voucher_no=12345 AND
client='EN' AND trans_type='AP'
UPDATE acrtrans SET status = 'N' WHERE status= 'Y' AND voucher_no=12345 AND
client='EN' AND trans_type='AR'
UPDATE acrtrans SET status = '' WHERE status= 'Y' AND voucher_no=12345 AND
client='EN' AND trans_type='GL'
UPDATE acrtrans SET status = 'X' WHERE status= 'Y' AND voucher_no=12345 AND
client='EN' AND trans_type='TX'
UPDATE acrtrans SET status = 'X' WHERE status= 'Y' AND voucher_no=12345 AND
client='EN' AND trans_type='TE'
```

Using CASE, we can write it like this:

```
UPDATE acrtrans SET status =
CASE trans_type
    WHEN 'AP' THEN 'N'
    WHEN 'AR' THEN 'N'
    WHEN 'GL' THEN ''
    WHEN 'TX' THEN 'X'
    WHEN 'TE' THEN 'X'
    ELSE 'Y'
END
WHERE status='Y' AND voucher_no=12345 AND client='EN'
```

SAMPLE

The script should be placed in a text file with asq-extension.

```
/* ASQL script sample */
PRINT '*** ASQL script sample ****'
/
/* abort script on the first error */
ON ERROR EXIT
/
/* drop the table if it exist */
IF EXISTS asqsample
/
    DROP TABLE asqsample
/
END
/
/* Create a table */
```

```

CREATE TABLE asqsample
(
    varchar_col varchar(256),
    number_col int1,
    money_col money,
    last_update date
)
/

/* CREATE unique index */
CREATE UNIQUE INDEX aliasqsample on asqsample (varchar_col,number_col)
/


/* Insert a row into the table */
INSERT INTO asqsample
(
    varchar_col,
    number_col,
    money_col,
    last_update
)
VALUES
(
    'test row 1',
    12,
    12.12,
    NOW
)
/


/* define a variable that will contain the agresso version */
DEFINE agresso_version char(50)
/


/* set a initial value for the variable */
SELECT 'Agresso version not found' INTO :agresso_version
/


/* If the table syssetup with column text2 exist, write the text2 column value into the variable
agresso_version*/
IF EXISTS syssetup text2 CONSTRAINT
/
    SELECT concat('Agresso version: ',text2) INTO :agresso_version
    FROM syssetup
    WHERE name = 'BASE_VERSION'
    /
END IF
/

```

```

/* Write agresso version to output*/
PRINT :agresso_version
/

/* define a variable*/
DEFINE agrdata char(50)
/


/* Run a MSSQL spesific query and insert tablespace for the table we created into variable agrdata*/
BEGIN sqlserver
/
    select distinct s.groupname into :agrdata
        from sysobjects t, sysindexes i, sysfilegroups s, syscolumns c
        where t.type = 'U'
            and t.id = c.id
            and t.id = i.id
            and (i.indid=0 OR i.indid=1)
            and i.groupid = s.groupid
            and t.name = 'asqsample'
    /
END
/


/* Script will continue if something fails in the rest of the script */
ON ERROR CONTINUE
/


/* Run a oracle spesific query and insert tablespace for the table we created into variable agrdata */
BEGIN oracle
/
    select tablespace_name into :agrdata
        from user_tables
        where table_name = UPPER('asqsample')
    /
END
/


/* Write tablespace to output */
PRINT :agrdata
/
PRINT '**** SCRIPT COMPLETE ****'
/

```

The script can be run with the command:

ASQL.exe -D<datasource> -F"<path to script>" -v -l"c:\asqllogfile.log"

56 News

Syntax

- LONGTEXT: the content of a LONGTEXT column can be maximum 32000 characters
- BLOBLENGTH(col_expr)
- CHARINDEX(string_expr1, string_expr2 [,start_location]) Searches expression2 for expression1 and returns its starting position if found. The search starts at start_location (optional).
- DATETIME2STR(date_expr) Converts a data value to a string on the format 'yyyymmdd hh:mi:ss'
- INT2STR(int_expr)
- REPLACE(string_expr, string_pattern, string_replacement) Replaces all occurrences of a specified string value with another string value
- STR2DATE(string_expr)
- SUBSTR(string_expr, start_expr, length_expr)
- TO_CHAR(expr) Converts an expression to string
- TO_COUNTER(expr) Converts an expression to int8
- TO_DATE(string_Expr) Converts a string of format 'yyyymmdd hh:mi:ss' to date (same as datetime2str)
- TO_GUID(string_expr)
- TO_INT(expr)
- TO_FLOAT(expr)
- TO_MONEY(expr)

And also the ranking functions

- DENSE_RANK () OVER ([< partition_by_clause >] < order_by_clause >)
Returns the rank of rows within the partition of a result set, without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the row in question.
- ROW_NUMBER () OVER ([<partition_by_clause>] <order_by_clause>)
Returns the sequential number of a row within a partition of a result set, starting at 1 for the first row in each partition.

ACCESS TO ORACLE FILES

Additional access permissions

When using Agresso Self Service and Agresso Web Services with Oracle, some additional access permissions must be set. The **Agresso Management Console** will grant **Read** access to the following directories:

- %ORAHOME%\bin
- %ORAHOME%\oracore\zoneinfo\
- %ORAHOME%\ocommon\nls\ADMIN\DATA
- %ORAHOME%\network\ADMIN (ldap.ora, sqlnet.ora)
- %ORAHOME%\rdbms\mesg

Users that needs access

The user running the *w3wp.exe* (application pool) process of the Agresso web application needs access. By default it is running as a Network service.

Troubleshooting

- If there is no access to *oci.dll* (in the *orahome\bin* directory), you will see the following message:
System.Data.OracleClient requires Oracle client software version 8.1.7 or greater.

- If you have access to *oci.dll* but are missing access to other Oracle files needed by the Oracle Call Interface, you will see this message:

Could not create an environment: OCIEnvCreate returned -1.

! Remember that it might be needed to delete temporary files and run iisreset after you have changed access to oracle files.

AGRESSO IN A TERMINAL SERVER (CITRIX) ENVIRONMENT

Introduction

The basic functionality of the Agresso Smart Client has been tested on Microsoft Windows 2008 with Citrix XenApp (Presentation Server) 5.0

We have found no particular problems specific to Agresso, but, since not everything has been tested thoroughly, there might be issues that require special attention (for example: use of local resources such as printers and scanners).

Installation options

Server recommendation

The requirements for running the Smart Client from a Terminal Server, are the same as for running the Smart Client on a PC.

Alternatives

You can either install the Agresso Smart Client locally on the server or use an existing centrally configured client.

Installing Smart Client (locally) from the DVD

Recommendation

We recommend that you install Agresso Smart Client from the console while the server is in [Install Mode](#) and no remote users are logged in.

When the client is installed and the data source has been configured using [Agresso Client Configuration](#), you should start up the Smart Client and check that it performs properly. If no problems occur, set the server back to [Execute mode](#).

Installing by using Centrally Configured Client

If you already have the Agresso components installed on a server, you can use a centrally configured client for running Agresso on the Terminal Server. A few issues must be considered, though:

- The Agresso client packages that are required on local PC's will also be required on the Terminal Server.
Note on SDS: It is not recommended to use [Agresso Software Distribution Server](#) (SDS) in combination with a Terminal Server. You must ensure that SDS does not try to install software on your Terminal Server when users start up the Smart Client.
- We recommend that the centrally configured client files are copied from your Agresso server and placed on a local drive on the Terminal Server before executed by users. This will ensure optimal performance. Running the centrally configured client from a shared folder on the network is also possible.

Agresso and Citrix Presentation Server

If your Terminal Server runs Citrix Presentation Server, you will have additional options for distributing the installed application to end users. Agresso supports Citrix Published Applications, and has been tested with the Access Suit WEB Interface (previously called Citrix NFuse) and Citrix Secure Gateway for access over Internet.

SDS AND WINDOWS FIREWALL (CLIENT)

Defaults

If the windows firewall is switched on SDS will be unable to perform operations on the client.

This can be solved either by

- turning off the Windows firewall, or
- giving access in the firewall.

Turning off Windows firewall

1. On local computers: Open [Control Panel | Windows Firewall](#) and de-select the radio button on the **General** tab

2. Using active directory policies: Go to [..\\administrative templates\\network\\network connections\\domain profile](#) and set the "Protect all network connections" to disabled.

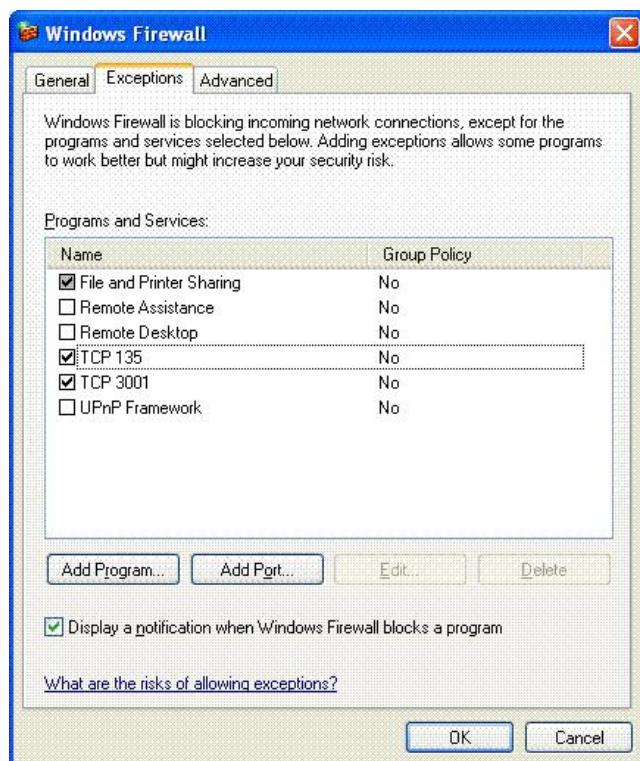
Giving access in the firewall

Ports: In order to give the necessary access, the following ports must be opened:

- TCP 3001,
- TCP 135 (RPC endpoint),
- UDP 137 (File And Print Sharing),
- UDP 138 (File And Print Sharing).

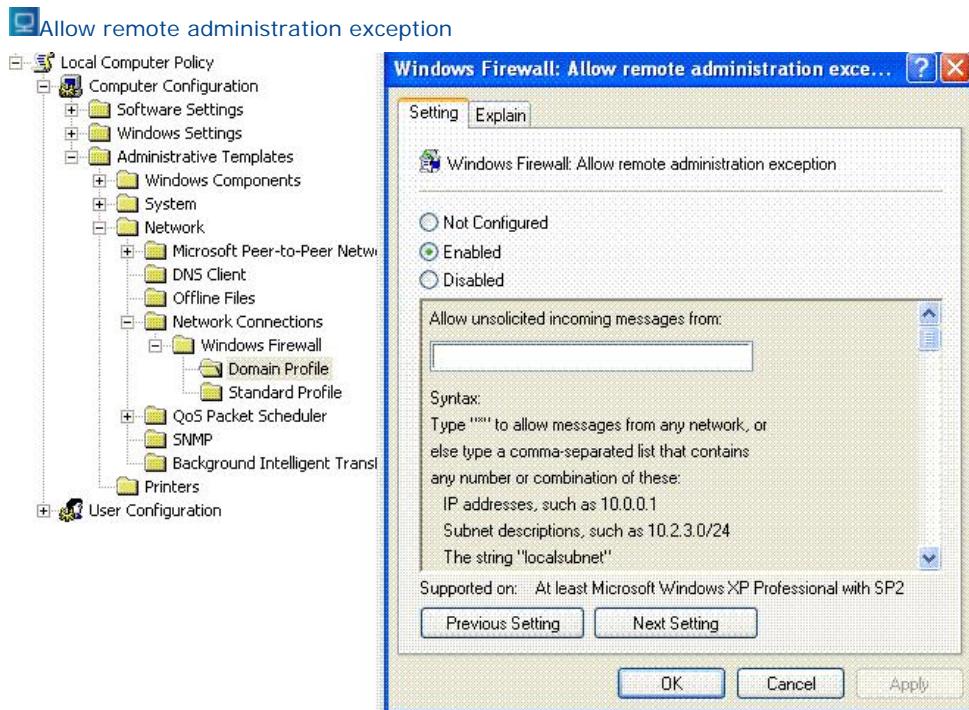
1. On local computers: Open [Control Panel | Windows Firewall](#) and check the (above mentioned) ports on the **Exceptions** tab.

 Firewall Exceptions on Local Computer



2. Using active directory policies: Go to [..\\administrative templates\\network\\network connections\\windows firewall\\domain profile](#), set "Define inbound port exception" to enabled, and add the ports listed above.

 On Windows XP, the policy "Windows Firewall: Allow remote administration exception" must be enabled in order to let SDS access clients with the windows firewall turned on. This policy enables RPC and DCOM access against the machine.



These settings are stored in the registry key `PrivilegedRpcServerPermission` on local machines. Default value: Not configured (disabled).

SDS AND WINDOWS FIREWALL (SERVER)

Defaults

On Windows 2003, the firewall was disabled by default, but on 2008 and 2008 R2 it's enabled and doesn't allow much traffic in. A consequence, SDS will not work.

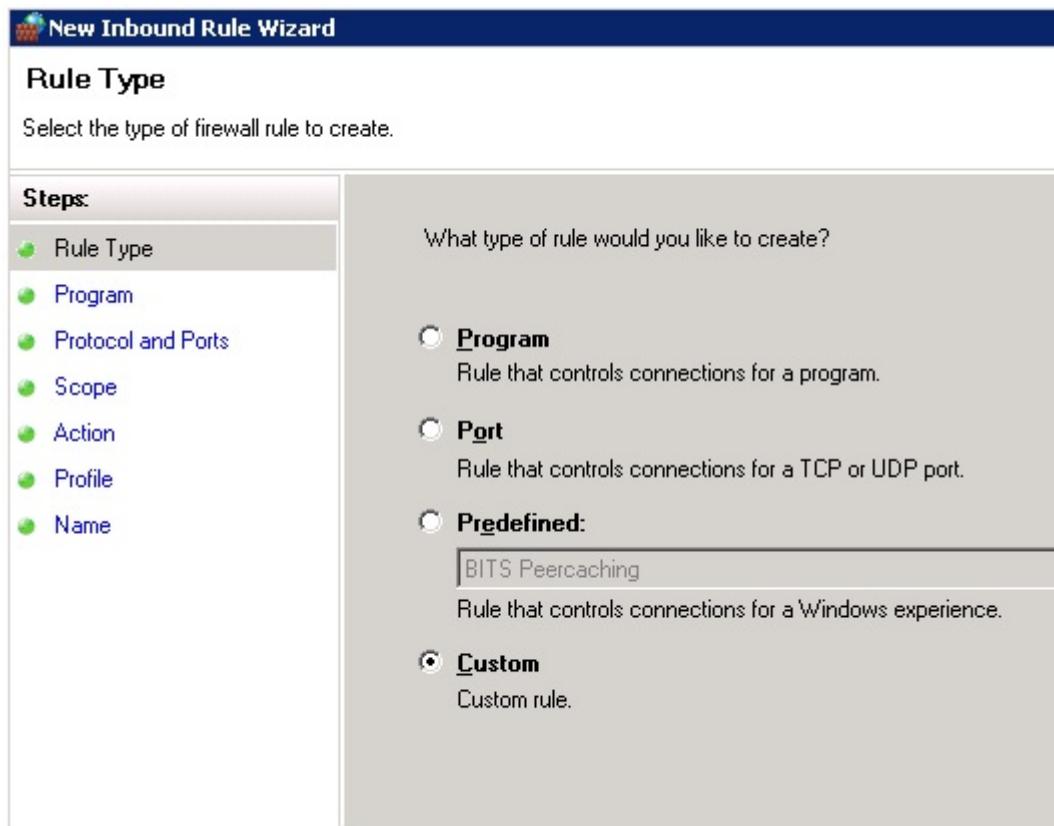
This can be solved either by

- turning off the Windows firewall, or
- giving access in the firewall.

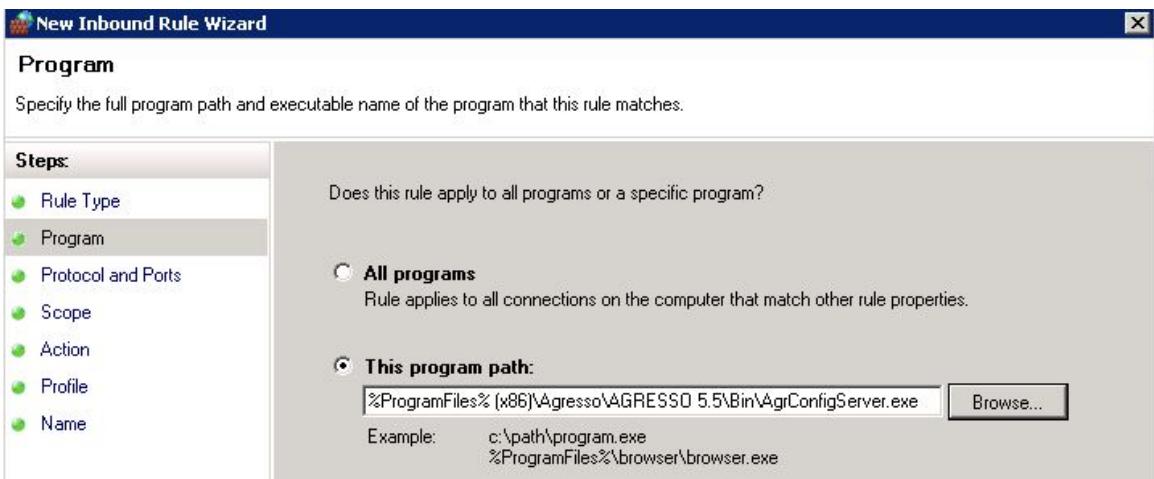
Giving access in the firewall

To set up the correct access for the SDS on the server side, go to the Windows Firewall with Advanced Security snap-in found in Administrative Tools. The default setting from Microsoft is that inbound connections are blocked unless there is a rule allowing traffic on that port/for that program. Outbound connections are allowed unless explicitly blocked.

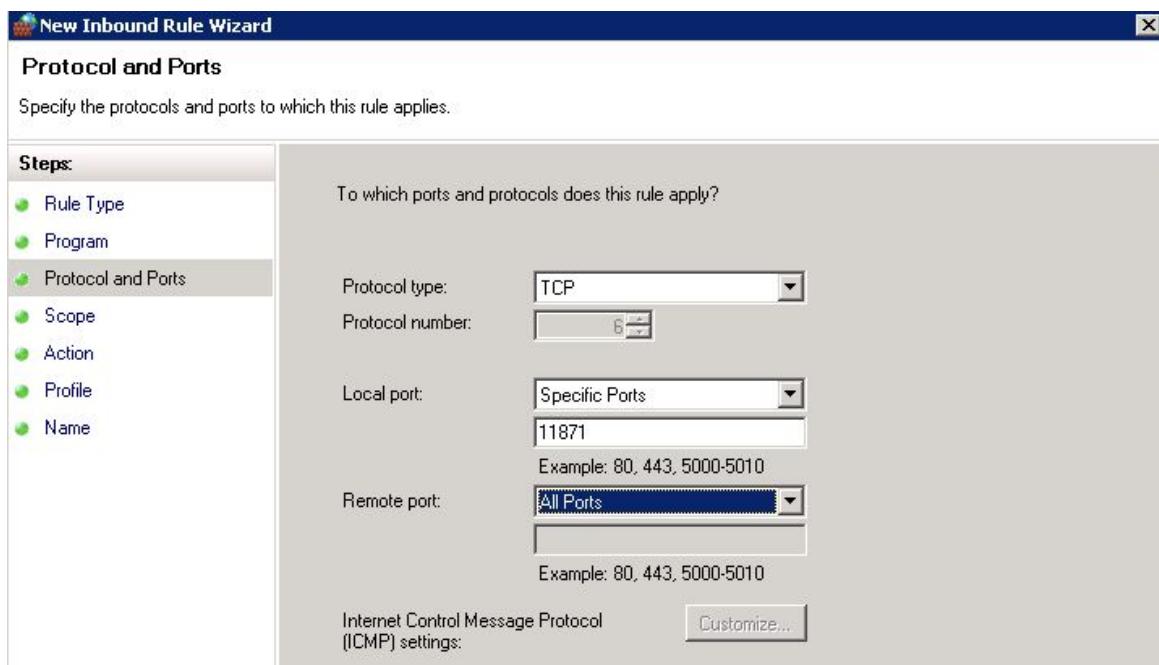
Right-click the "Inbound Rules" node and select "New Rule". Choose Custom "Rule Type"



Click next, and browse to AgrConfigServer.exe located in the Agresso bin-folder.

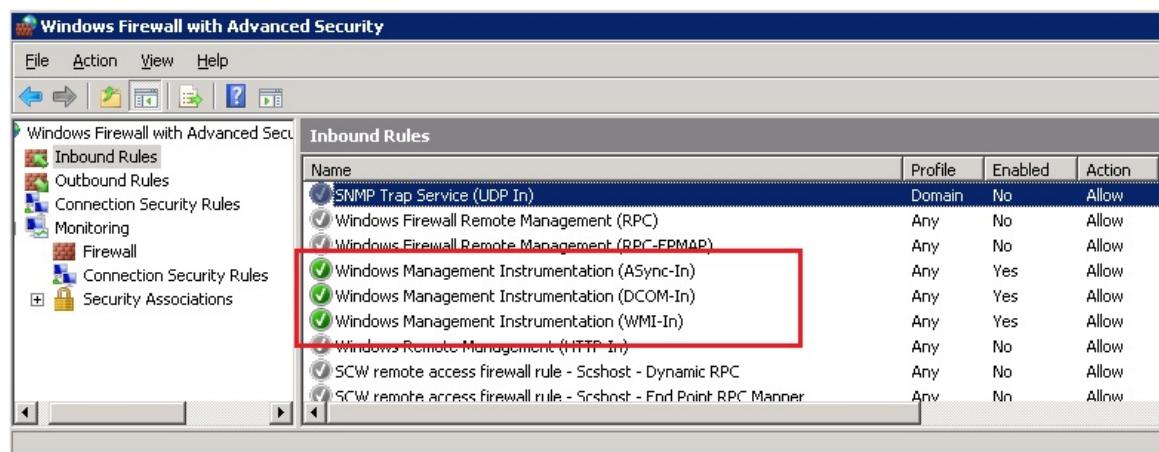


Click next, and select port. The default used by the Software Distribution Service is port 11871, but this can be changes in AMC and agresso32.ini if needed.



Complete the wizard (there is no need to make any changes to the default configuration on the subsequent steps in the wizard).

Three existing inbound rules that are disabled by default needs to be enabled.



The windows firewall is now configured for Agresso SDS.

ORACLE SYNTAX FOR CORRELATED UPDATES

5.5 improvement

The lack of join syntax in traditional Oracle updates has been the cause of much tuning effort. Agresso 5.5 is using a new syntax for updates with join that seems to solve this problem. Testing has shown a significant speed improvement for many server jobs.

With this syntax Oracle is performing updates similar to SQL Server both when it comes to performance and function.

It is possible to have an ambiguous replace. If the join gives two or more different records that match the one to be updated, you will get no error and thus not know which value was used for the update. This is consistent with the behavior of SQL Server.

However, the normal business logic in Agresso prevents this kind of duplicates.

Sample Code

Version 5.4:

```
UPDATE HGL1933
SET ( HGL1933.description1 ) =
  (SELECT a.description
   FROM agldescription a
   WHERE a.attribute_id = 'A0' AND a.dim_value = HGL1933.dim1 AND
         a.client = 'EN' AND a.language = 'EN' )
WHERE EXISTS (
  SELECT a.description
  FROM agldescription a
  WHERE a.attribute_id = 'A0' AND a.dim_value = HGL1933.dim1 AND
        a.client = 'EN' AND a.language = 'EN' )
```

Version 5.5:

```
UPDATE /*+ bypass_ujvc*/
  (SELECT a.description adescription , h.description hdescription
   FROM Hagr56_2_GL120003 h , agldescription a
   WHERE a.attribute_id = h.attribute_id
         AND a.dim_value = h.dim_value
         AND a.client = 'EN'
         AND a.language= 'EN' )
  SET hdescription = adescription
```

The hint used in the new syntax /*+ bypass_ujvc*/ is needed.

Oracle 11.2 (The bypass_ujvc hint is not supported in Oracle 11.2) :

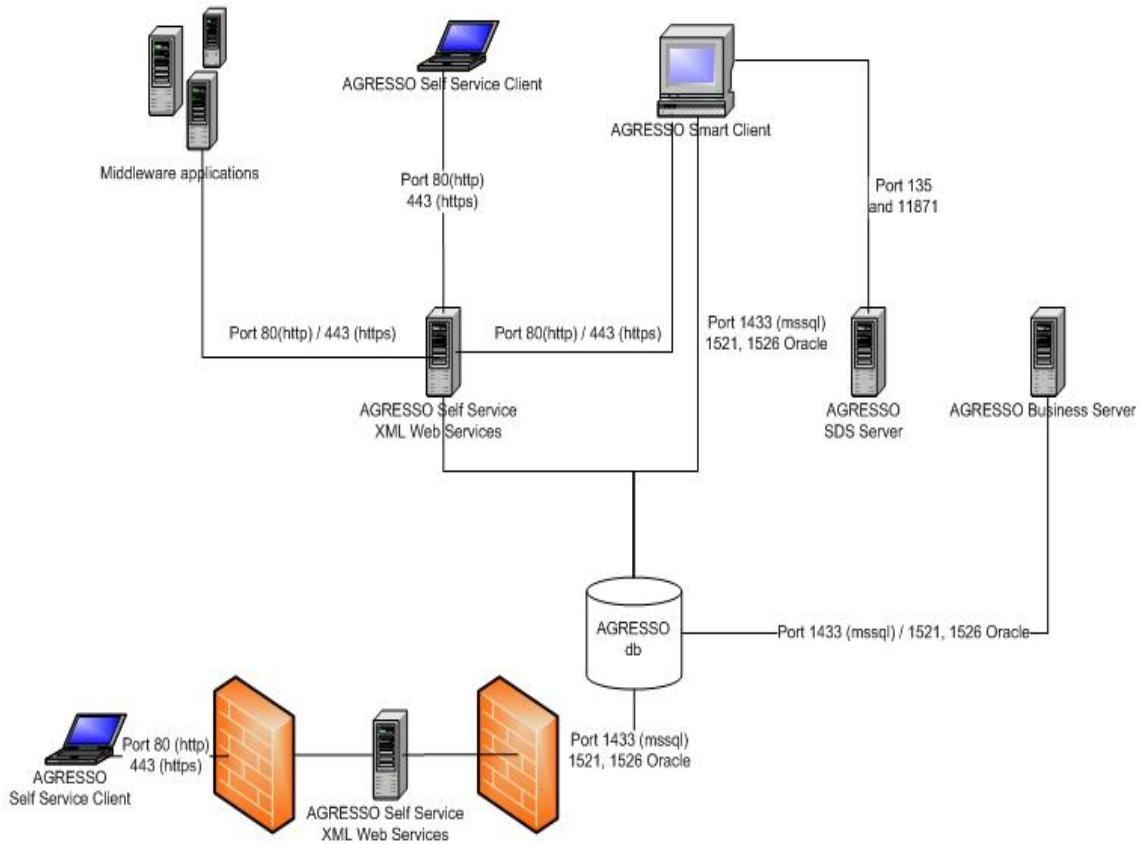
```
MERGE INTO Hagr56_GL120003 h
  USING (SELECT a.description AS description, h.rowid xzfd_rid
          FROM agldescription a, Hagr56_GL120003 h
         WHERE a.attribute_id = h.attribute_id
               AND a.dim_value = h.dim_value
               AND a.client = 'EN'
               AND a.language = 'EN' ) xzfd_t
    ON (xzfd_t.xzfd_rid = h.rowid)
  WHEN MATCHED THEN
    UPDATE SET h.description = xzfd_t.description
```

Typical Agresso network topology

Overview

The diagram outlines the communication between server and client components in a typical Agresso installation:

Default ports used within the Agresso infrastructure



CERTIFICATE REVOCATION CHECK

Introduction

The Agresso binaries are signed with a certificate from *Verisign*.

The default Windows behavior is that each time an Agresso executable or a Web application is launched, the Certificate Revocation List (CRL) needs to be verified in order to check if the certificate has been revoked by the Certificate Authority.

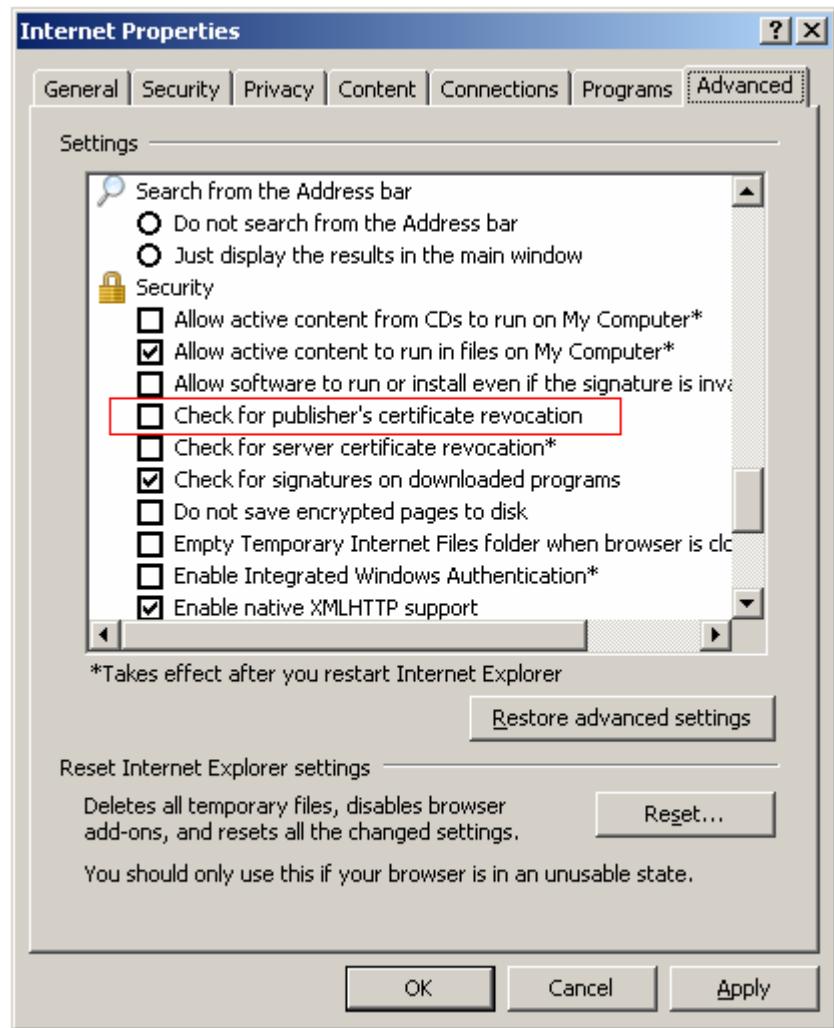
The CRL needs to be downloaded from Verisign, so the different applications needs access to the Internet - or you can turn off the CRL check.

If the application is unable to download the CRL, it will wait for a time out which will add about 35 seconds to the start-up time of a process.

Certificate Revocation Check on the Smart Client

In a locked down environment, the best solution is to give all clients access to <http://crl.verisign.com>. Alternatively, the CRL check can be turned off.

This can be done on the **Advanced tab** in **Internet Options**, found in the **Control Panel**.



Certificate Revocation Check on the Web Server

Depending on the server configuration, starting an Agresso Web application (Self Service client or a Web service) shouldn't take more than ten seconds if the CRL check succeeds (or is turned off). If the CRL check fails it may take up to a minute. The CRL check is done each time the process is started, not each time it receives a request.

Configure a proxy for server processes

By default, the Web applications are run under the local Network Service account. This user will normally not have access to the Internet in a corporate environment because it won't have a proxy configured. One way of configuring a proxy for server processes is to use *ProxyCfg.exe* which ships with Windows.

Example syntax: `proxycfg -p <proxy_server> "<local>"`

This is the most common use for *ProxyCfg.exe*. The command specifies that both HTTP and HTTPS servers are accessed through the proxy server named `<proxy_server>`, except for host names that do not contain a period.

Reference: See <http://msdn2.microsoft.com/en-us/library/aa384069.aspx> for more details about *ProxyCfg.exe*.

Turn off CRL check

It is also possible to turn off the CRL check for system accounts through the registry. If the Web Application is running as the default user, **Network Service**, you can change the **State** value under:

`HKEY_USERS\S-1-5-20\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing`

The default value is **23c00**. Changing it to **23e00** will turn off the CRL check. Note that this is a per-user setting, so this will only affect the Network Service account.

Certificate Revocation Check on the Business Server

If the AGRESSO Business Server service is configured to run as a user without access to the Internet (for instance Local System or a normal domain user that don't have access to the outside world), it will take approximately 35 seconds longer to run a report.

Normal user settings

To configure settings for a normal user, just log in as that user and configure either the proxy settings or disable the CRL check through the Internet Options dialog in the Control Panel.

System User settings

If the service is running as **Local System**, either configure the proxy settings with *ProxyCfg.exe* (see above) or turn the CRL check off by changing the **State** value found under:

HKEY_USERS\S-1-5-18\Software\Microsoft\Windows\CurrentVersion\WinTrust\Trust Providers\Software Publishing

The default value is **23c00**. Changing it to **23e00** will turn off the CRL check.

How to check for CRL check problems.

General

Any packet sniffer can be used to check for CRL check problems, but the easiest way is to use **Process Explorer**. (see <http://www.microsoft.com/technet/sysinternals/ProcessesAndThreads/ProcessExplorer.mspx>).

Start **Process Explorer**, right-click the relevant process and look at the **TCP/IP** tab. If the process is unable to contact Verisign there will be a request with state **SYN_SENT**. The process will not continue until this request times out.

Smart client

To check for CRL problems on the Smart Client, right-click the *Agresso.exe* process.

Web server

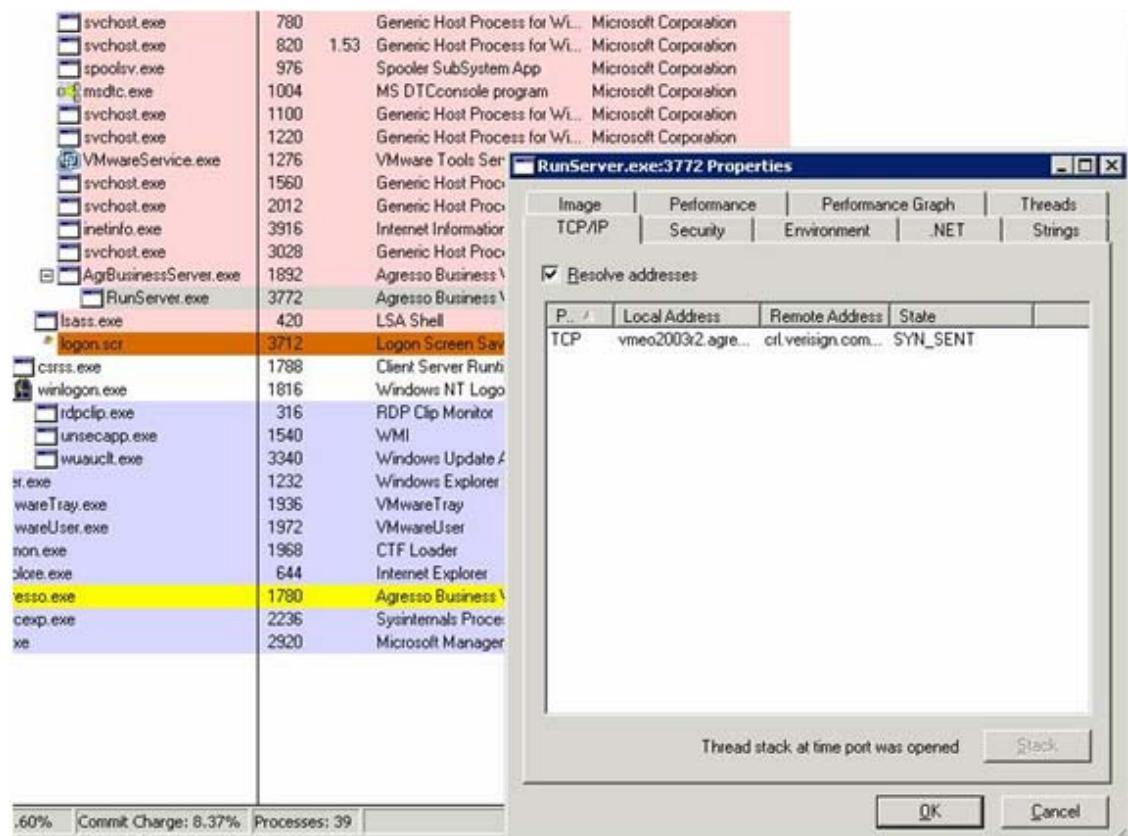
On the Web Server, check the *w3wp.exe* process that is running the Agresso web application. To find the correct process, check the **Command Line** on the **Image** tab. It will look something like this for the Self Service application:

```
c:\windows\system32\inetsrv\w3wp.exe -a \\.\pipe\iisipm18fbce65-579e-405e-a855-3ded323903f6 -t 20 -ap "AgressoAppPool"
```

Business server

On the Business Server, check one of the *RunServer.exe* processes.

Diagram: The example below shows the situation when *RunServer.exe* fails to contact Verisign:



Agresso Database Tools

TOOLS OVERVIEW

Three tools

There are three main tools supplied by Agresso to manipulate the database during an installation, and later during maintenance and database upgrades:

- Agresso Copy
- Agresso SQL
- AgrIndex

Agresso Copy

[Agresso Copy](#) is used to copy table definitions and data between databases and database management systems (RDBMSs).

There are an Agresso Copy program for each supported database. If using Oracle database, the name is [copyora.exe](#), while the MS SQL alternative is named [copyms.exe](#).

! In this Appendix we are discussing the *copy* command in detail and how it used from the command line. For information on how to use the *copy* feature from the Management Console, please refer to the [Configuration](#) section.

Agresso SQL

[Agresso SQL](#) is a general purpose SQL interpreter, which is available for all of Agresso's platforms and databases. The Agresso SQL program is called [asql.exe](#) for all databases.

AgrIndex

[AgrIndex](#) ([AgrIndex.exe](#)) is a console application used to re-generate indexes for Agresso tables.

Agresso Copy

AGRESSO COPY

Functionality

Specialised tools

The Agresso database copy programs, **copyora** and **copyms** and are standalone programs used to load tables into a new Agresso database and to move tables from one database to another (for example during an upgrade).

To copy data between two Oracle databases, you may also use the Oracle exp/imp utilities. However, in order to move data between different database management systems, you need to use the Agresso copy programs.

Furthermore, in some cases these programs are easier to use than the native tools. For instance, the Agresso copy programs will delete all tables by default before they are re-created and the data is loaded. Using native Oracle import, you must delete the tables manually before running import.

files.lst

The copy programs are used directly from the command prompt. They will copy table definitions and indexes to and from ASCII files. A file named **files.lst** controls which tables are copied. Each table copied out of a database has its table definition, index definitions and data placed in a separate file. The **files.lst** file contains the names of all these files.

By default, the copy programs drops and re-creates all tables in the control file list. Make absolutely sure that your **files.lst** is correct before issuing a **copy in** command. The **files.lst** in the current directory is always used, unless **Agresso Copy** is told otherwise through command line parameters.

Views

The copy programs can also create views in the database. The view definitions must be in the table **aagview** in the same database. Use the **-v** flag during **copy in** to create views.

Note: During **copy out** the **-v** flag has no relevant effect.

Examples

Task: You have copied out a complete database, but want to copy only the tables relevant to the Agresso Fixed Assets module back in (afa* tables).

Solution: You edit **files.lst**, and removes all lines not starting with **afa**.

Task: Copy in all the tables for the Contract module (acn* tables).

Solution: If you have saved the original **files.lst**, you can edit this file and make a new **files.lst** as described above.

You can also type **DIR /B acn*asq > files.lst**. This command will create a new **files.lst** containing only acn* entries.

Syntax

General

The most common syntax for **Agresso Copy** for the different databases is:

MS SQL: **copyms.exe -d<direction> -U<user> -P<passwd> -S<odbc datasource> [-z -v] [<tables>]**

Oracle: **copyora.exe -d<direction> -U<user> -P<passwd> -D<database> -S<tnsname> -z -v [<tables>]**

Example: If you shall copy all Agresso tables from MSSQL, database **agresso**, to Oracle, you write:

copyms.exe -dout -Uagresso -Pagresso -S<odbc datasource> a%

copyora.exe -din -Uagresso -Pagresso -Dagresso -S<tnsname> -z -v -a1000

For **copy in**, no table names should be specified. The tables (files) listed in **files.lst** will be copied.

List all options

In order to get an overview of all available copy options, enter the following at the command prompt:

```
copyms -? | more
```

The -a parameter (Oracle)

The **-a** parameter is very important for Oracle. If this parameter is set to a large number, e.g. 5000, the program will copy 5000 rows at a time, consuming lots of memory, but speeding up the copy process. The **-a** option is only relevant for Oracle databases.

The -z parameter

The **-z** parameter tells the database to create procedures and functions from the *asysddl* and *aagddl* tables.

Null values

In general, Agresso tables are defined with the **not null** and **default value** options for every column. Allowing null values in a table is not acceptable in Agresso, and the application will fail.

AGRESSO COPY SYNTAX

There is one copy program for each database. If using Oracle database, the name is *copyora.exe*, and if using the MS SQL database, the program is called *copyms.exe*. These programs have the same functionality. Every table copied in or out has its own data file associated with it (990+ files when copying a full Agresso database). The data file is an ASCII file and can be edited by the user (e.g. change table name/column name/location, remove/change index, etc.). Tables with blob will create one directory for each table as the name of the table, and all blob data will be in that directory.

Agresso Copy is a part of the application server software in Agresso Business World. The program is started from AGRESSO_EXE (\agresso\server), in command prompt window.

The copy programs have a lot of parameters, but the most common syntax for Oracle is:

```
copyora -dout -U<user> -P<password> -S<dbserver> <table names>
copyora -din -U<user> -P<password> -S<dbserver> [-A] [-v] [-z]
```

The **-D** parameter is not necessary for Oracle. For Oracle the **-S** is used for TNS-names. SQL Server use the **-S** parameter for an ODBC data source.

The parameters can be set in any order, but the table names must be entered after the last parameter. Notice that no table names are usually entered when data is copied into a database (table names are retrieved from the files.lst file). It is possible to supply a list of table names for "copy in" as well, in which case only these tables will be loaded and not all the tables listed in files.lst.

You can list as many table names as you want (restricted only by OS). Table names are separated by white space. Valid wild cards are (1) underscore (match exactly one arbitrary character), and (2) percentage sign (match any number of arbitrary characters).

! In Agresso Business World there is a new set of tables used by Agresso Invoice Manager. These tables use different database settings, and must be copied separately. These tables start with a "c".

Data files

Regardless of which database the data was copied out from, the data files all look the same. They contain the name and location of the table, column definitions, etc., all divided into sections.

The name of the data file consists of the first eight characters of the table name, a punctuation mark and a three digit number. If agltransact is the 32nd table to be copied out, the data file will be called agltrans.031 (starts from 000).

Table section

```
[TABLE]
name=acsgroup
location=agrstatic
```

This first section has two parameters; name and location.

The location parameter is only used if the -T flag is set (see description of parameters to copy programs). If location=default the parameter is also ignored.

Columns section

```
[COLUMNS]
account,char,,8,,0,1,,
apar_gr_id,char,,2,,0,1,,
....
```

Each line in the columns section consists of five values:

- name (beware of case sensitive databases)
- type (char, varchar, date, bool, int1, int2, int, float, money, raw)
- length
- null values
- default value

The length value is only used for char or varchar columns. The other lengths can be ignored. When copying data from an Informix table, a money column may have a length of 8000+. This is normal.

If the length is larger than 255 (or = 16), it will be ignored for the following columns:

- authorize (which will be given length 1500)
- col_09 (which will be given length 700)
- col_10 (which will be given length 500)
- col_11 (which will be given length 500)
- date_string (which will be given length 366)
- info (which will be given length 366)
- q_param (which will be given length 1500)
- q_where (which will be given length 400)
- query (which will be given length 1600)
- text (which will be given length 720)

If columns are *not null* and all columns have default values (apart from special tables in Agresso Invoice Manager):

- char, varchar: default "";
- bool, int1, int2, int: default 0
- money, float: default 0.0
- date: default Jan 1 1900, 00:00:00

Index section

```
[INDEXES]
aiacsgroup1, agrindex,1,apar_gr_id, apar_type, client
```

The index section consists of one line for each index. The first value is the index name and must be less than 18 characters. The next value is the physical location of the index. The third value is a unique flag, and the rest of the values are the names of the columns included in the index.

Parameter section

```
[PARAMETERS]
col_sep=,
line_sep=[EOLN]
string_quote="
```

This section informs the copy program how the data is stored. Do not change anything here.

Data section

[DATA]

```
"2010","3","P",0,"NO",0.000000,"NOK",0,"Random suppliers",0,  
" ",1,"NO",19980312 01:07:00," ",0,"1580"," ",0," ",0,"30",0,"sysno",
```

If deleting everything in front of this section, you will have a clean data file which can be used in other programs (e.g. Excel). If you change something in the data section, make sure that the changes do not conflict with the column definitions. Note that there is a comma at the end of each line of data.

files.lst

There is a special file created by Copy out, called *files.lst*. This is the first file read by Copy in. It contains a list of all the data files in the directory.

If you have a directory containing data files for all the tables in an Agresso database, but only need three tables, you can change *files.lst* so only the data files for these three tables are listed here. The copy program will then ignore the rest of the data files in the directory. This can also be done by using a "dir" command to create a new *files.lst*:

```
dir /b a*.* > files.lst
```

Make sure that there are no empty lines or special characters in *files.lst*. This may stop the program prematurely.

AGRESSO COPY PARAMETERS

Mandatory parameters

Parameter descriptions

When starting the program with no parameters, you will get an overview of the syntax for Agresso Copy. Remember that there should be no space between a parameter identifier and its value.

The following table gives complete list of the mandatory parameters. Note that the parameters must come in the sequence listed below.

Parameter	Description	Example
-d	States copy direction, in to database or out.	<code>-din</code> <code>-dout</code>
-U	Login name. Specifies the login name for the user. When using MS SQL as the database, make sure that the login name is using SQL Server Authentication.	<code>-Uadminuser</code>
-P	The password for the user logging in. Make sure the login specified by the <code>-U</code> parameter is using SQL Server authentication, and that the password is correct.	<code>-PmyPassw1</code>
-D	MS SQL only: The login name specified with the <code>-U</code> parameter always have a default database it connects to. The <code>-D</code> parameter is only used if the login name has access to more than one database, and you want to copy from/to a different database than the default.	<code>-Dagr55</code>
-S	States logical server name Oracle: The database logical server name. Copyora needs a valid net service name to connect to the database. MS SQL: The ODBC data source.	<code>-Sagr55</code>
<tables>	List of tables to be copied. You can list	<code>a%</code> (will copy all tables)

	as many tables as you want (limited by the OS). You can also use % (any strings) and _ (any character).	beginning with a)
--	--	-------------------

Example

The following command will

Optional parameters

The following are a complete list of the optional parameters with syntax and description:

» Append

Syntax: `-A`

This parameter instructs the copy program not to re-create tables before data is read into it. Make sure that the table definition in the database corresponds to the data file. If not, it results in an error message for each row.

» Array size (batch size)

Syntax: `-a<size>`

This parameter is only used by Oracle, not MS SQL. The purpose of the "-a" parameter is to tell the copy program how many lines it should insert at a time. The larger the value, the faster the program works. However, if this value is too large, you will eventually run out of log or memory. Default is 100. When this value is set to zero, all rows are inserted one at a time (slow).

» Maximum number of blob files per directory

Syntax: `-B<size>`

Blobs in a table are copied into separate files called `i000000000000000n.raw`, where "n" is a row counter. Depending on the disk and the OS, there is a limit to how many files a directory can hold. The copy program will spread the files into separate directories if there are more than 32000 rows in the table. 32000 is the default, but can be changed using this variable.

» Where flag

Syntax: `-b<0/1>`

This flag is used in conjunction with the "-q" flag. If a "where" statement is used when copying out data, you can decide whether or not you want it stored in the data file with the data. It can then be used to replace data when copying into a database. Valid values are:

- 0: Do not add the "where" statement to the data file when copying out, or, do not use the "where" statement when copying in. This is the same as not setting it at all.
- 1: When copying out, the "where" statement is added to the data file. When copying in, data in the table where the "where" statement is true, is deleted. The data in the data file is then added. The table is not re-created, so datatypes and lengths of columns must match the datafile.

» Column separator

Syntax: `-c<char>`

This parameter is used when you want to change the column separator. It is only used when copying data out. If you are copying Agresso tables, using this parameter is not recommended. In addition to a character, [TAB] and [EOLN] are valid values.

» Default values

Syntax: `-e0`

All the columns are created without default values. If this parameter is not set, character columns will have an empty string as default, numeric columns will get "0" as default, and date columns will get "Jan 1 1900" as default.

» File directory

Syntax: `-f<path>`

Unless this parameter is used, all the data files are created/read from the current directory (the directory you are calling the program from). If the directory does not exist, the program will try to create it. A path can be given relative to the current directory.

» Display only

Syntax: `-g`

Only used with parameter "-din". The create and drop commands are echoed to "stdout". No tables are dropped or created and no data is read. If the result from the copy command is redirected to a file, the file can later be run from within interactive SQL. Views are created regardless of this parameter.

» Name of list file**Syntax:** `-h<file name>`

Unless this parameter is used, a file called files.lst will be created in the same directory where your data files are. This file is a listing of all files which are going to be read. Notice that the file cannot contain path names or any information other than file names (e.g. comments, etc.).

» Index flag**Syntax:** `-i[0/1]`

You can ignore the index section by using "-i0". If you only want to ignore the unique flag, use "-i1".

» Index location**Syntax:** `-I[<location>]`

If this parameter is used without any value, the index location parameter in the data file is used. If you supply a location, all indexes are created in this location (dbspace, tablespace or segment).

» Progress indicator**Syntax:** `-j<rows>`

The program will print a dot to stout for every <rows> row copied.

» Line separator**Syntax:** `-l<char>`

This parameter should not be used. The default line separator is newline and should not be changed.

» Identity column**Syntax:** `-m`

Used by MS SQL only. When this parameter is set, the identity column (agrtid) is not added when the table is created.

» Null values**Syntax:** `-n`

Use this flag if you want to allow "null values" in the tables. Agresso standard tables should not be allowed to have "null values". However, special tables for Agresso Invoice Manager must allow "null values".

» Column names**Syntax:** `-o<col_names>`

Only tables containing a column name in the list of column names are retrieved.

» Original datatype**Syntax:** `-O`

Datatype text in MS SQL Server will be converted to varchar2(4000) in Oracle. This should only be used when copying special tables for Agresso Invoice Manager.

» Where statement**Syntax:** `-q<string>`

You can limit the amount of data copied out by using this parameter. Supply an SQL "where" clause enclosed in double quotes.

» Automatic ANSI to OEM conversion check**Syntax:** `-Q`

If you run the SQL Server Client Network Utility, you will find a setting called "Automatic ANSI to OEM conversion" under the "DB-Library information" tab. If this is checked (turned ON), it may cause problems when copying text containing characters that are not part of the English alphabet (like some Scandinavian characters). The copy program will automatically check this setting to make sure it is OFF. Use "-Q" if you do not want to perform this check.

Trigger warnings

Syntax: `-r`

If the table being copied contains a trigger, the definition is printed to stdout. When copying into a database, the trigger is not re-created.

Raw (blob) data file name

Syntax: `-R<column_name>`

When tables containing blobs are copied, every blob is stored in separate files called `i00xxxx.raw`. If your table contains `file_name` information for every blob, you can use these when the blob is copied out. Give the column name containing the file information as parameter `-R`. If the `file_name` is blank, or there are duplicates, the default `i00xxxx.xxx` is used (it will only keep the file extension).

String quote

Syntax: `-s<char>`

In the data file, all strings are enclosed in double quotes. If you want to use single quotes or something else (maybe no quotes at all), use this parameter.

Table location

Syntax: `-T[<location>]`

If this parameter is used without any value, the location parameter in the data file is used. If you supply a location, all tables are read into this location (dbspace, tablespace or segment).

Unicode

Syntax: `-u`

Tables is by default created using "char" and "varchar". The parameter "`-u`" results in tables created with types "nchar" and "nvchar".

View

Syntax: `-v`

Creates database views. Only used with "copy in". View definitions are read from the table `aagview`.

Extended copy

Syntax: `-x`

Invoice Manager tables must be copied with special parameters (see section [Parameters used with Invoice Manager tables](#)) that are not used for Agresso core tables. Tables starting with "c" is treated as Invoice Manager tables. If for some reason you do not want to treat tables starting with "c" as special tables, use `-x0`.

NATIVE COPY PROGRAMS

Native programs

All the database servers can copy data to and from text files, but there is no standard way of doing this, and there is no standard format for the files. Usually you have to do the copy operation in several steps.

For the database servers we are using, the native programs listed below can be used. These programs have many parameters and some need different types of parameter files.

In order to copy data between Agresso databases more easily, Agresso Copy was made.

Oracle

Programs: exp, imp, sqlloader.

Advantages: Fast, can handle one or more tables, very powerful ('r;everything' is possible).

Disadvantages: Does not drop tables, Difficult to read/write 'r;foreign files'

MS SQL Server

Program: bcp.

Advantages: Fast, easy to use.

Disadvantages: Does not drop tables. One table at a time. A special database option must be set.

Agresso SQL

AGRESSO SQL PROGRAM OVERVIEW

Agresso SQL Program (asql.exe)

This program executes SQL statements. You can use either Agresso SQL syntax or native database syntax. The statements are stored in an ASCII file with `.asq` file extension. This file is referred to by the parameter (-F) at the command line when running the program.

! *This section is discussing detailed information on ASQL program and syntax. For information on how to use Agresso SQL Script from the Management Console, please refer to the document [Agresso SQL Scripts](#) in the Configuration section.*

Connection parameters

The following is a description of the input parameters and the Agresso SQL syntax. Some of the input parameters, called the *connection parameters*, vary from platform to platform. Parameters common to all platforms are called *input parameters*.

`>>-D<Agresso Data Source>`

The -D parameter refers to an Agresso data source. Prior to running asql.exe, a data source must be set up. This can be done using the Agresso Management Console. The -D parameter should not be used if you use the parameters below for alternative connection.

Alternative connection parameters

`>>-U<Database User>, -P<User Password> and -D<Database Name>`

The parameters below should only be used in special circumstances where you are unable to use an Agresso Data Source.

! Agresso environment variables cannot be used unless you connect with an Agresso Data Source.

Under special circumstances asql is used when there is no Agresso Data Source, e.g. Agresso Budget (Budwin). If you are not using an Agresso Data Source, there are several parameters you can use instead of -D:

ODBC/MS Sql Server: These parameters are optional. If not given, database will be the default database connected to the login name.

! If you change default database on the odbc data source (-S below), this will have no effect here. It will still be the default database from the login name that is used.

Oracle: These parameters are not in use in an Oracle environment.

`>>-S<Native Source>`

ODBC/MS Sql Server: ODBC data source (-D<database name> is optional).

Oracle: Oracle Net Service Name (-D<database name> is not used).

`>>-I<Interface>`

Specify RDBMS type (default value is ODBC):

- ODBC
- ORACLE
- NATIVE (If you select NATIVE, you must specify Agresso database driver with d<agrdb.dll>)

`>>-d<Db driver>`

Only used in conjunction with the parameter INATIVE. In that case, you must specify and Agresso database driver, e.g. agrodbc.dll or agrora.dll.

General input parameters

The following general input parameters are available:

`>>Path to asq files, -f`

This parameter is not required. If the asq files are not in the current directory, you can use this parameter to tell asql.exe where the files are stored. Normally this parameter is used in conjunction with the -h parameter.

This parameter will override the environment variable AGRESSO_SCRIPT.

If the -h parameter is used and the list file contains path names, this parameter is ignored.

>> Name of list file, -h

A list file is a file containing the names of one or more asq files. When using the -h parameter, asql.exe will run all the asq files listed in the list file. The files can be entered in the list file with either full or relative path. If the path is given, it will override the -f parameter and AGRESSO_SCRIPT.

This parameter will override the -F parameter (name of asq file).

>> Name of asq file, -F

The name of the file containing the SQL commands to be executed. This file will normally have the file extension asq. You can enter the full path name. If no path name is entered, asql.exe will first look in the directory specified in the environment variable AGRESSO_SCRIPT. If this environment variable is not defined, it will look in the current directory.

This parameter is overridden by the -h parameter giving the name of a list file.

>> Exit flag, -x

This parameter is only used when running more than one asq file (i.e. -h parameter is used). It is used in conjunction with the "on error exit" command. The "on error exit" command's scope is only the current file. Thus, if an error occurs after this command, asql.exe will skip the rest of the current file, but continue with the next one. However, if you want asql.exe to cancel all files when an error occur, this parameter is used.

>> Verbose flag, -v[+]

All commands and information are printed to a log file. This file is called asql.log, and is found in the directory specified in the AGRESSO_LOG environment variable. If this environment variable is not defined, the log file will be created in the current directory.

Every time you run asql.exe with the parameter -v, the current log file is replaced with the new one. If you don't want to delete the old file, but append to it, use the + switch.

Environment variables

The following environment variables are used by asql.exe:

>> AGRESSO_IMPORT

This environment variable is used by the copy statement and points to the directory where the data file is:

>> AGRESSO_LOG

The directory (full path) to where the log file is created. If not set, current directory is used.

>> AGRESSO_SCRIPT

The directory (full path) to where the asq file(s) are found. If not set, current directory is used.

Agresso SQL Syntax

AGRESSO SQL SYNTAX OVERVIEW

ASQL

Agresso SQL is an SQL variant tailored to facilitate Agresso specific database operations. ASQL is translated into the native SQL syntax of the host database when executing.

Native Database SQL can be enabled by prefixing the SQL with the [DATABASE keyword](#) or by using [database blocks](#).

ASQL features include:

- Rich set of functions
- Database independent data types
- CREATE TABLE FROM SELECT with any host database

- Importing data from text files

ASQL does not support:

- Outer joins
- NULL values

Data definition language (DDL)

The following commands are used to create and remove tables and views:

Command	Description
CREATE TABLE	Creates a new table in the database
CREATE TABLE AS	Creates a new table in the database based on succeeding query
DROP TABLE	Removes a table from the database. The table definition will disappear
CREATE VIEW AS	Creates a view from parts of one or more tables
DROP VIEW	Removes a view from the database. The table definition will disappear

Data manipulation language (DML)

The following commands are used to retrieve, insert, delete and update rows in a table.

Command	Description
SELECT	Retrieves data (rows) from a table
INSERT	Inserts new data (rows) into a table
UPDATE	Changes existing data (rows) in a table
DELETE	Deletes data (rows) from a table
MERGE [INTO]	Merges content from two tables
COPY	Copies rows between a table and text file

Index commands

An index is a table of pointers. Each row in the indexes points to a corresponding row in a data table.
Unique indexes are used to ensure that a table contains no duplicate row.

Command	Description
CREATE UNIQUE INDEX	A column or combination of columns in a table that uniquely identifies each row in the table. (primary index)
CREATE INDEX	Defines an index on a table (secondary index)
DROP INDEX	Removes an index

Comments

Comments can be added anywhere in an asq-file, even inside statements. However, they can not be used inside strings. Comments are in the /* */ form and will be ignored when reading the file.

```
>>Sample code
UPDATE mydescription m
  FROM description d
    SET m.description = d.description
  /*
      , m.amount = fmulf(m.amount,abs(d.mult))*1.1
  */
  WHERE m.account = d.account
```

```
AND m.client = d.client
```

The amount column will not be updated.

Log messages

Writelog

You can use the WRITELOG keyword to improve the messages in the log file.

» Sample code

```
Example A:  
insert into mytable (col1, col2, col3)  
select col1, col2, 0 from myothertable  
  
Example B:  
writelog Inserting into :tablename  
insert into :tablename (col1, col2, col3)  
select col1, col2, 0 from myothertable  
/
```

In the example A, you get the following in the log file:
13:02:01 2 > Execute statement

In example B, you get:
13:02:01 2 > Inserting into mytable

Notice that you can use variables directly in your log message.

Print

The PRINT statement can be used to put text into your log file. The string can contain variables.

» Sample code

```
print :count items found when looking  
/
```

DATA TYPES AND FUNCTIONS

Data types

The following data types can be used to define columns in the tables that you are going to create in the database. The CONVERT function only recognises da of the types CHAR | DATE | FLOAT | INT | MONEY.

Type	Length	Description
char	1–255	Fixed length character string. Length normally <= 16
vchar	1–4000	Variable length character string
bool	N/A	Normally used as a flag. Valid values are 0 or 1
int1	N/A	Small integers. Valid values from -255 to 255
int2	N/A	Medium size integers. Valid values from -32768 to 32768
int8	N/A	Large integer.

int	N/A	Normal integers. Valid values from -2147483648 to 2147483648
float	N/A	A float value. Valid values depend on database server
money	N/A	A money value. Legal values depend on database server. Do not expect a precision of more than 16
date	N/A	A date/date time value
guid	N/A	A global unique identifier. A new identifier can be created with GETGUID()
longtext	N/A	long text columns. varchar(max) on mssql, clob on oracle. Max length: 32000 chars.
raw	N/A	Used to store binary large objects (BLOBS)

Aggregate functions

The arithmetic operators + , - , * , \ can be used.

Function	Description
COUNT(*)	Returns the number of rows in the specified table
COUNT(all distinct expr)	Returns the number of (distinct) non-null values in a column
SUM(all distinct expr)	Adds up the values in a specified column
AVG	Returns the average of all the values in the specified column
MAX	Returns the maximum value that occurs in the specified column
MIN	Returns the minimum value that occurs in the specified column

Numeric functions

Function	Description
ABS(n)	Returns the absolute value of a numeric expression (n)
ROUND(n,x)	n rounded to x dp
MOD(n,b)	n modulus b
SQRT(n)	square root of n
fdiv(expr1,expr2)	Converts expr1 and expr2 to float and divides. Type of the return value is float
fditm(expr1,expr2)	Same as fdiv, but type of the return value is money
fmul(expr1,expr2)	Converts expr1 and expr2 to float and multiplies. Type of the return value is float
fmulm(expr1,expr2)	Same as fmul, but type of the return value is money
mod(expr1,expr2)	Returns the integer result of expr1 modulo expr2
min(col)	Returns the lowest value
max(col)	Returns the highest value
power(expr1, expr2)	Returns expr1^expr2
sum(col)	Returns the sum of column values
sqrt(expr)	Returns square root of expr

sample

```
PRINT '*** ASQL script sample - math functions ***'
```

```
/
```

```

ON ERROR EXIT
/
IF EXISTS asqsamples_math
/
DROP TABLE asqsamples_math
/
END
/
CREATE TABLE asqsamples_math
(
    description varchar(256),
    abs_number int1,
    round_number int1,
    mod_number int1,
    sqrt_number int1,
    fdiv_number float,
    fdivm_money money
)
/
INSERT INTO asqsamples_math
(
    description,
    abs_number,
    round_number,
    mod_number,
    sqrt_number,
    fdiv_number,
    fdivm_money
)
VALUES
(
    'Test sys functions',
    abs(-1.23),
    round(123.569932, 1),
    mod(5,2),
    sqrt(8),
    fdiv(8,3),
    fdivm(8,3)
)
/

```

Conversion

Function	Description
convert(from,to,expr)	Converts a value from one datatype to another. The first two parameters can have one of the following values: char, int, float, money, or date. (In Oracle, money, int and float are the same type.)
int2str(integervalue)	Returns <i>integervalue</i> as a string

to_counter(expression)	Returns <i>expression</i> as an int8.
to_float(expression)	Returns <i>expression</i> as a float.
to_int(expression)	Returns <i>expression</i> as an int.
to_money(expression)	Returns <i>expression</i> as money.

String functions

Function	Description
bloblength(col_expr)	Returns the number of bytes in a blob column.
charindex(searchstring, searchfor[, startindex])	Returns index of first character of <i>searchfor</i> in <i>searchstring</i> . -1 if not found. Optionally, you can start the search at <i>startindex</i> .
concat(expr1,expr2)	Concatenates one string to another
length(expr)	Returns length of expression
left(expr, n)	Returns the leftmost n characters of a string
lpad(c,length,expr)	Pads the leftmost string expr with c characters. The length of result string is length Note: Padding with blanks will not work since all result strings are trimmed.
lshift(expr,n)	Shifts value of the string expr n characters to the left
lower(expr)	Returns the input string (expr) in lowercase
replace(originalstring, stringtoreplace, replacewith)	Replaces all occurrences of <i>stringtoreplace</i> in <i>originalstring</i> with <i>replacewith</i> .
right(expr,n)	Returns the rightmost n characters of a string
rpad(c,length,expr)	Pads the string expr with c characters. The length of result string is length. Note: Padding with blanks will not work since all result strings are trimmed.
rtrim(expr)	Returns the string expr without trailing blanks
space(n)	Returns a string containing n spaces (max 1800)
squeeze(expr)	Returns the string expr without leading and trailing blanks
substr(expr, startindex, numchars)	Returns a substring of <i>expr</i> starting at <i>startindex</i> and includes <i>numchars</i> characters.
to_char(expression)	Returns the expression as a string. Ref. ToString() i C#.
upper(expr)	Returns the input string (expr) in uppercase

» sample

```
PRINT '*** ASQL script sample - string functions ***'
```

```
/  
ON ERROR EXIT  
/  
IF EXISTS asqsamples_string  
/  
DROP TABLE asqsamples_string  
/  
END  
/
```

```

CREATE TABLE asqsamples_string (
    description varchar(256),
    concat_string varchar(25),
    length_int int1,
    left_string varchar(25),
    lpad_string varchar(25),
    lshift_string varchar(25),
    lower_string varchar(25),
    max_string varchar(25),
    right_string varchar(25),
    space_string varchar(25),
    squeeze_string varchar(25),
    substr_string varchar(25)
)
/
INSERT INTO asqsamples_string
(
    description,
    concat_string,
    length_int,
    left_string,
    lpad_string,
    lshift_string,
    lower_string,
    right_string,
    space_string,
    squeeze_string,
    substr_string
)
VALUES
(
    'Test sys functions',
    concat('text1','text2'),
    length('length of this string'),
    left('this is returned. This is not', 16),
    lpad('a', 10, '0123'),
    lshift('0123456789', 4),
    lower('MAKE ME LOWER'),
    right('return to forever', 4),
    space(10),
    squeeze(' do not fill in the blanks '),
    substr('0123456789', 2, 4)
)
/

```

Misc functions

Function	Description
convert(from,to,expr)	Converts a value from one datatype to another. The first two

	<p>parameters can have one of the following values: CHAR, INT, FLOAT, MONEY or DATE (In Oracle, MONEY, INT and FLOAT are the same type)</p> <p>The following conversions gives unpredictable results: DATE ?? ? INT, FLOAT or MONEY</p>
ifnull(expr,expr)	Returns the second expression if the first returns null
ascii(character)	Returns the asci code for a character. ascii('A') returns 65

» sample

```

PRINT '*** ASQL script sample - sys functions ***'
/
ON ERROR EXIT
/
IF EXISTS asqsamples_sys
/
DROP TABLE asqsamples_sys
/
END
/
CREATE TABLE asqsamples_sys (
    description varchar(256),
    convert_char varchar(15),
    convert_float float,
    ifnulll_char varchar(15),
    int2str_char varchar(15)
)
/
INSERT INTO asqsamples_sys
(
description,
convert_char,
convert_float,
ifnulll_char,
int2str_char
)
VALUES
(
'Test sys functions',
convert(float, char, 12.3456),
convert(char, float, '12.3456'),
ifnull(NULL, 'was null'),
int2str(15)
)
/

```

Date functions

Function	Description
AGRDBTODAY()	Returns date for current day

AGRDBNOW()	Returns the current date and time
cts2day(string)	Converts a string to date and truncates to day (time part is set to 00:00:00)
date2iso(date)	Returns a string 'yymmdd' (substracted from date)
datediff(date1,date2)	Returns a floating point number (date1 - date2). The integer portion is number of days
datepart(unit,date)	Returns an integer containing the specified part of the date. Unit can have one of the following values (date parts): DAY, HOUR, MIN, MONTH, QUARTER, SEC, WEEK or YEAR
datetime2str(adate)	Returns the date value <i>adate</i> to a string ("yyyymmdd hh:mi:ss")
datetrunc(unit,date)	Returns a date value that represents the input date truncated to the level of granularity expressed in parameter unit. Unit can have one of the following values (date parts): DAY, MONTH or YEAR
dayadd(n,date)	Adds n days to input date and returns the new date. Use negative number of days (n) to subtract
getdate()	Returns current date
secadd()	Adds n seconds to input date and returns the new date.
monthadd(n,date)	Adds n months to input date and returns the new date. Use negative number of months (n) to subtract
date2str(date)	Converts a date to string
iso2date('yymmdd')	Converts a string 'yymmdd' to a date
str2date(stringdate)	Returns a date from a string in the format 'yyyymmdd hh:mi:ss' (as the next)
to_date((stringdate)	Returns a date from a string in the format 'yyyymmdd hh:mi:ss' (as above)
ts2day(date)	Truncates date to day (time part is set to 00:00:00)

 sample

```

PRINT '*** ASQL script sample - date functions ***'
/
ON ERROR EXIT
/
IF EXISTS asqsamples_date
/
DROP TABLE asqsamples_date
/
END
/
CREATE TABLE asqsamples_date (
    description varchar(256),
    agrdbnow_date date,
    getdate_date date,
    now_date date,
    cts2day_date date,
    date2iso_date date,
    datediff_float float,
    datepart_int int2,
```

```

datepart_date date,
dayadd_date date,
monthadd_date date,
ts2day_date date,
agrdbtoday_date date,
date2str_str varchar(25)
)
/
INSERT INTO asqsamples_date
(
description,
agrdbnow_date,
getdate_date,
now_date,
cts2day_date,
date2iso_date,
datediff_float,
datepart_int,
datepart_date,
dayadd_date,
monthadd_date,
ts2day_date,
agrdbtoday_date,
date2str_str
)
VALUES
(
'Test date functions',
AGRDBNOW(),
GETDATE(),
NOW,
CTS2DAY('12/12/79'),
date2iso(to_date('20080731 01:02:03')),
datediff(to_date('20080731 01:02:03'), to_date('20080730 01:02:03')), /* will return 1 - one day*/
datepart(sec, to_date('20080731 01:02:03')), /* will return 3 - three seconds */
datetrunc(day, to_date('20080731 01:02:03')), /* will return 2008, rest of the day will be */
dayadd(2, to_date('20081231 01:02:03')),
monthadd(2, to_date('20081231 01:02:03')),
ts2day(to_date('20080731 01:02:03')),
AGRDBTODAY(),
date2str(to_date('20080731 01:02:03'))
)
/

```

Guid functions

Function	Description
getguid()	Returns a new guid.
guid2str()	Returns a string representation of a guid.

to_guid(guidasstring)	Returns a guid based on a formatted string value.
-----------------------	---

```

>>sample
/* ASQL script sample - Test guid functions*/

on error exit
/

if exists asqsamples_guids
/


DROP TABLE asqsamples_guids
/
end
/
CREATE TABLE asqsamples_guids (
    description  varchar(256),
    my_guid      guid
)
/
INSERT INTO asqsamples_guids (description,my_guid) VALUES ('Test getguid function', {fn GETGUID()})
/

```

Ranking functions

Asql supports two ranking functions:

```
DENSE_RANK() OVER([< partition_by_clause >] <order_by_clause >)
```

Returns the rank of rows within the partition of a result set, without any gaps in the ranking. The rank of a row is one plus the number of distinct ranks that come before the row in question.

```
ROW_NUMBER() OVER ([<partition_by_clause>] <order_by_clause> )
```

Returns the sequential number of a row within a partition of a result set, starting at 1 for the first row in each partition

Macros

Function	Description
MAX_DATE	Highest acceptable date (20991231 23:59:59)
MIN_DATE	Lowest acceptable date (19000101 00:00:01)
NO	0 (FALSE)
NOW	Current date and time (date type - not char)
TODAY	Current date (at 00:00:00) (date type - not char)
YES	1 (TRUE)

NATIVE DATABASE SQL

Options

If you want to use native SQL syntax within ASQL, you have two options:

- use the keyword **database** before the actual SQL statement. The Agresso parser will keep the statement as it is, and not try to translate it from ASQL to native code.

- use database blocks to delimit a set of SQL statements which the Agresso parser shall ignore. A database block is identified by the keywords begin and end:

DATABASE keyword syntax example

If you need to create a sequence in ASQL, this is not supported by ASQL. You write the following instead:

```
DATABASE CREATE SEQUENCE mysec
```

where the CREATE SEQUENCE statement will be resulting (parsed) SQL statement.

Database Block syntax example

Database server type

A database block is delimited by the keywords BEGIN ... END. Immediately after BEGIN, you must identify the database server type. This type must correspond to the actual server(!) and can be one of the following:

- ORACLE
- SQLSERVER
- SQLSERVER7

Example

The following examples illustrates the use of database blocks, without actually displaying any native code:

Oracle:

```
begin oracle
    <native SQL>
end
```

MS SQL:

```
begin sqlserver
    <native SQLSERVER SQL>
end
```

IF ... BLOCKS

IF [NOT] ... END IF

IF blocks are typically used to create tables that don't exist, drop them if they do exist, or alter them if a column is missing or defined wrong. But you can also use an IF block in a general way.

Syntax

There are two main syntax variants:

Table manipulation syntax:

```
if [not] exists <table> [<column>] [constraint]
<statements>
end if
```

General syntax:

```

if [not] SELECT < ... >
<statements>
end if

```

Table manipulation

When using EXISTS, The IF statement will check if the table or column exists, and run the statements if the condition is true. You can also use a select statement, instead of an object.

! *Checking on constraints should only be used for SQL Server. You can use the key word "constraint" if you want to enter the "if" clause if a constraint exists on the column, but you don't know the name.*

Examples

```

if not exists testtab
/
create table testtab (col1 int, col2 char(10))
/
end if
/

...
if exists testtab col1 constraint
/
alter table testtab drop col1 constraint
/
end if
/

...
if exists testtab col1
/
database alter table testtab drop col1
/
end if
/

```

General use of IF ... blocks

When used without EXISTS and <table>, you use the IF to determine actions on basis of what the SELECT returns (null or zero rows = false)

Example

```

IF NOT SELECT id from tab where id = 3
/
INSERT INTO tab (id, col) VALUES (3, 'value')
/
END
/

```

If there is no rows with id = 3, the statement will insert a new row into the table tab.

VARIABLES

ASQL variables

Scope

Agresso SQL allows you to use variables. These variables must be defined outside database blocks, but can be used inside the same blocks.

When a variable is defined, it will remain defined throughout the program. This means that you can not define the same variable in more than one file, if you want to run these files together (using the -h parameter).

Define variables

Variables can be used almost anywhere in a statement. You define them using the DEFINE statement and assign values using a SELECT statement. You refer to the variables using a colon in front of the name (eg. :myvar).

You can give your variable any name except those that are Agresso SQL keywords. The name should not be in quotations.

Data types

Valid data types are explained below:

Type	Description
char	Normal string with max length 255.
ident	ASQL-specific string with max length 40. An ident variable will always be assigned a string value. When referred to, however, it will return the object (table, column etc) identified by the assigned string value. See example below.
int	Standard int.
float	Standard float.
date	Standard date.

Example

In the following example, we define 3 variables and copies values from 3 columns in the last row in `org_table` into them. The column `testtab` refers to a table, and we put the value (table name) into the ident variable `mytable`.

The final update statement then updates the table referred to by `mytable`.

```
define char description(60)
/
define ident mytable(32)
/
define date last_update
/
select testtab, description, last_update
    into :mytable, :description, :last_update
    from org_table
/
update :mytable
    set description = :description
    where last_update < :last_update
/
```

A note on the INTO clause: You assign variable values in the INTO clause. The first element in the SELECT list corresponds to the first variable in the INTO clause, the second element corresponds to the second variable, etc. It does not affect the value assignments if there are more SELECTed elements than variables.

TRANSACTION AND ERROR HANDLING

Transactions

You can group INSERT and UPDATE statements together. This is done by transaction statements. Statements inside a transaction can be viewed as one single statement. It either succeeds or fails.

Make sure you do not have Data Definition Statements (create or drop statements) inside a transaction.

BEGIN

The BEGIN statement marks the beginning of a transaction.

ABORT

The ABORT statements abort a transaction. All statements executed after the last BEGIN transaction statement are undone. You cannot call ABORT transaction without having called BEGIN transaction.

END

The END statement marks the end of a transaction. All the statements called since the BEGIN transaction statement are now committed to the database.

 Try to keep transactions short. As long as you are inside a transaction, you keep locks on the tables you are manipulating. This leads to low concurrence and can also lead to deadlocks.

Error handling

When a statement does not execute because of an error, the default behaviour is to report the error message and then continue with the next statement. This behaviour is controlled by the ON ERROR statement (see also the input parameter exit flag).

There are two options:

- **On error exit:** When an error occurs, the program will either exit or skip to the next .asq file (see input parameter exit flag). If you are inside a transaction, all statements (since begin transaction) are rolled back.
- **On error cont:** This is the default behaviour (if no EN ERROR statement is used). When an error occurs, you continue with the next statement.

DATA DEFINITION LANGUAGE (DDL)

DDL

A data definition language (DDL) supports the definition or declaration of database objects. Data definition statements are used to define and maintain database objects. In Agresso SQL these statements are introduced by one of the following:

CREATE (table, index, view),

ALTER,

DROP.

Note: Data definition statements should NOT be inside transactions. If the database server allow this, make sure the transaction is inside a database block.

Create table

You create a table by specifying a table name and column names with data types.

Note: In an ASQL Query series the table will automatically be dropped before it is created. Also all columns will be created to not allow NULL values, and with default values

General syntax

The general syntax is as follows:

```
create table <tab_name>
  {as <select statement> |
  (<col_name> <data_type>[<(length)>], ...)}
  [tabspace = <location>]
```

Syntax elements

The various syntax elements are explained below:

Element	Description
<tab_name>	Name of the table you are creating.
<select statement>	A normal select statement in Agresso SQL syntax.
<col_name>	The name of a column.
<data_type>	The data type of the column. See Data Types and Functions for an overview of valid values for type and length.
<location>	The tabspace clause is used when you want to put the table on a location (tablespace, segment, dbspace) other than the default

Example - Create table

```
CREATE TABLE department
  (depname CHAR(25),
  depno INT,
  amount MONEY,
  responsible INT,
  date_from DATE,
  date_to DATE,
  flag INT,
  company CHAR(2),
  hours FLOAT)
```

Example - Create table from another table

Two examples:

```
CREATE TABLE testdepartment AS
SELECT depno, depname, company
FROM department
WHERE type = 'test'

...
CREATE TABLE testdepartment AS
SELECT depno, depname, company,
SPACE(50) AS deptmanager,
CONVERT(INT,FLOAT,0) AS hours
FROM department
WHERE type = 'test'
```

Restrictions

Number of columns: The maximum number of columns is 250.

Bytes in a row: Some database servers restrict the total number of bytes for a row to less than 2KB.

Long columns: There should never be more than one raw column in a table, or more than one char/vchar column longer than 255 bytes.

ORDER BY not allowed: When creating tables with a SELECT statement, the ORDER BY clause can not be used.

Create index

An index is a table of pointers. Each row in the index points to a corresponding row in the table. You gain speed by accessing records through an index. Although there is a specific screen for maintaining indexes in Agresso it does not include temporary tables.

General syntax

The general syntax is as follows:

```
create [unique] index <ind_name>
on <tab_name> (<col_name>, ...)
[[tabspace = <location>]]
```

Syntax elements

The various syntax elements are explained below:

Element	Description
unique	Adds a unique constraint to the table. Two rows where all the columns listed in the index have the same value, is not allowed
<ind_name>	Specifies the name of the index
<tab_name>	Specifies the name of the table
<col_name>	Specifies a column name. You should try and limit the number of columns in the index. Agresso does not allow more than 10.
<location>	The tabspace clause is used when you want to put the index on a location (tablespace, segment, dbspace) other than the default

Example

```
CREATE UNIQUE INDEX aidepartment1 ON department (depno, company)
CREATE INDEX aidepartment2 ON department (responsible, client)
```

Create view

A view is a virtual table that does not have any existence in its own right, but is derived from one or more underlying base tables. Views can be created at any time.

General syntax

```
create view <view_name> [(<col_name>, ...)]
as <select_statement>
```

Syntax elements

The various syntax elements are explained below:

Element	Description
<view_name>	Specifies the view name.
<col_name>	Specifies a column name. The view should not have more than 250 columns.
<select_statement>	A select statement with some limitations. You can not use an order by clause and you can not use set operators (like union).

Alter table

("Old" ASQL, not recommended for PL version 2.2 and later. Use [Modify table](#) instead. See below)

It is possible to change the table definition. For example, you can change the width of a column, delete or add columns, or drop a constraint.

Note: Dropping a constraint should only be used against SQLServer.

General syntax

The general syntax is as follows:

```
alter table <table> {add|drop} <column> [{constraint|<constraint>}]
[<description>]
```

Syntax elements

The various syntax elements are explained below:

Element	Description
<table>	The name of the table.
<column>	Specifies the name of the column.
<constraint>	The name of the constraint. If you do not know the name of the constraint, but want to drop it anyway, use the keyword "constraint".
<description>	This can only be a legal Agresso column type specification (bool, int1, int2, int, int8, float, money, varchar, char and date)

Examples

Two examples:

```
alter table testtab add col3 'varchar(50)'
/
alter table testtab drop col3 constraint
/
...
Alter table atsproject modify project char(12)
Alter table atsproject delete project
Alter table atsproject add project char(12)
```

Modify table

(Recommended for Platform 2.2 and later! Corresponds to "standard" Alter table in MS SQL and Oracle)

Modify versus Alter: No quotes around type expressions

The most important, general change from ALTER TABLE, is that there shall be **no quotes** around type expressions.

The old statement

```
ALTER TABLE ADD newcolumn 'varchar(50)'
```

shall be written

```
MODIFY TABLE add newcolumn varchar(50)
```

Formal syntax

The general syntax for MODIFY TABLE is as follows:

```
MODIFY TABLE <table>
{
    ADD [WITH CHECK] <column> <type_expression> [default {<str>|<num>}]
        [, <column> <type_expression>...]   |
    DROP <column> {CONSTRAINT | <constraint>}   |
    MODIFY <column> <type_expression> [default {<str>|<num>}]
}
```

Syntax elements

The various syntax elements are explained below:

Element	Description
<table>	The name of the table.
<column>	Specifies the name of the column.

<code><constraint></code>	The name of the constraint. If you do not know the name of the constraint, but want to drop it anyway, use the keyword "constraint".
---------------------------------	--

General example

```

MODIFY TABLE MyTestTable
    ADD WITH CHECK newcol150 varchar(50), newcol100 varchar(100)

```

Note: using WITH CHECK: When using the WITH CHECK option, the command will check against the system tables – for all columns listed in the command – whether the column exists or not. This can slow down performance.

If a column already exists, a message is written to the log, and the column is ignored.

Modifying columns

Modifying a column on a table is done very differently in Oracle and MS SQL.

MS SQL: In MS SQL, it is done in three steps:

1. drop any constraints on the column,
2. change the column type,
3. add a default value.

Oracle: In Oracle, everything has to be done in one statement.

Restrictions: MODIFY TABLE in ASQL allows you to write identical MODIFY <column name> statements for both Oracle and MS SQL. It is important, however, that you are aware of the following restrictions:

- You cannot modify a column in MS SQL without dropping the constraint first (you can in Oracle).
- You cannot modify a column in MS SQL if it is part of an index (you can in Oracle).
- You cannot modify a column in Oracle, if the column has any values (you can in MS SQL).
- The column must already have a NOT NULL constraint.

Drop tables, view and indices

Drop table or view

The general syntax for deleting a table or a view, is as follows:::

```

drop {table | view} <name>

```

where <name> identifies the table or view to drop.

Example: `DROP testdepartment`

Note: The `DROP` table <name> command will destroy both the content of the table and the table definition.

Temporary tables: It is not necessary to run `DROP` table <name> for temporary tables. Agresso will by default drop them at the end of the process.

Drop index

The syntax for deleting an index, is as follows:::

```

drop <index_name> on <table_name>

```

Note: The '`on <table_name>`' part is not mandatory. If you omit it, however, performance can be real slow!.

DATA MANIPULATION LANGUAGE (DML)

DML

A data manipulation language (DML) supports the manipulation and processing of database objects. The following commands are described:

- `SELECT`,
- `INSERT`,
- `DELETE`,
- `UPDATE`.

- MERGE.

Select

Syntax

The general syntax for SELECT is as follows:

```
SELECT [DISTINCT] column1, column2 ...
FROM table
WHERE <condition>
[GROUP BY column1, column2, ...]
[HAVING <condition>]
[ORDER BY column1, column2, ...]
```

Asterisk: Instead of specifying columns, you can use an asterisk (*), meaning all columns:

```
SELECT * FROM table ...
```

The DISTINCT keyword

When the DISTINCT keyword is used, duplicate rows are ignored. Only distinct (different) rows, compared to all previously retrieved, will be added to the result table.

The WHERE <condition> clause

The WHERE <condition> clause restricts the returned rows to those matching the <condition>, and where the condition in general corresponds to a specific value in one or more columns.

Examples: Typical examples of the WHERE <condition> as given below:

```
WHERE cust_name <> 'agresso'
```

...

```
WHERE cust_name <> 'visvas' and income > 10000
```

Comparison operators: The following comparison operators ar used in ASQL:

Operator	Description
=	equal to
!=	not equal to
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to

Logical operators: The following logical operators ar used in ASQL:

Operator	Description and examples
[NOT] operator [ALL ANY]	???
IS [NOT] NULL	Example: <pre>WHERE tab_name IS NOT NULL</pre>
[NOT] EXISTS	Example: <pre>WHERE exists (SELECT * from agltransact WHERE <voucher_date> TODAY)</pre>
BETWEEN	Example: <pre>WHERE date BETWEEN date_from AND date_to</pre>
[NOT] IN	Example:

	<code>WHERE name IN ('per' "kari,)</code>
[NOT] LIKE	Use to compare two strings for a partial match: Example: <code>WHERE name LIKE 'A%'</code>
AND	Validates two conditions, and both must be true. Example: <code>WHERE cust_name = 'MyCust%' AND income >= 500000</code>
OR	Validates two conditions, and one must be true. Example: <code>WHERE cust_name = 'MyCust%' OR income >= 500000</code>
NOT	Negates a condition value. The expression is true if the condition is false. Example: <code>WHERE NOT (cust_name = 'MyCust%')</code>

ORDER BY

ORDER BY is used to sort the query result in either ascending (default) or descending order. The columns in the ORDER BY-statement must appear in the SELECT list.

Example - order result in ascending order:

```
SELECT depno, depname, responsible
FROM department
ORDER BY responsible, depno
```

Example - order result in descending order

To order the result in descending order, you use the keyword DESC:

```
SELECT depno, depname, responsible
FROM department
ORDER BY depno DESC
```

GROUP BY

GROUP BY gathers rows into groups and sorts the groups. Used with aggregate functions, for example SUM, you can get a nice result that is almost ready for printing.

The columns in the ORDER BY statement must appear in the SELECT list.

Example - order by employee group:

```
SELECT empl_group, empl_no, description, SUM(amount)
FROM payroll
GROUP BY empl_group
```

HAVING

HAVING is used to eliminate groups that do not meet the search criteria.????

Example: In the following example ????

```
SELECT empl_no, name, AVG(salary)
FROM employees
GROUP BY empl_no, name
HAVING AVG(salary) > 180000
```

UNION

UNION (with additional keywords DISTINCT or ALL) is used if you have two tables and you need to include data from both tables.

Example: In the example below, we will get a result consisting of

```

SELECT second_name,
first_name
FROM prospect
UNION
SELECT second_name,
first_name
FROM customer

```

Using aliases

When writing join queries, aliases are very useful. The amount of typing can be cut down by using aliases (or correlation names). An alias is a short name that stands for a table name.

Example: The query:

```

SELECT employees.name, department.depname
FROM employees, department
WHERE employees.depno = department.depno
AND employees.client = department.client

```

can be simplified to:

```

SELECT a.name, b.depname
FROM employees a, department b
WHERE a.depno = b.depno
AND a.client = b.client

```

Cartesian Product

The simplest join is a two table SELECT in which there are no WHERE statement. Every row of the first table is joined with every row of the second table. The result is the Cartesian product of the two source tables.

Example:

```

SELECT a.empno, a.name, a.depno, b.depname
FROM employees a, department b

```

Equal Join

By applying constraints to the join with a WHERE clause, unwanted rows can be filtered out. The WHERE clause should contain conditions specifying that the value in the first table must be equal to the value of the corresponding column in the second table. To avoid Cartesian product all the columns in the unique index should be included in the WHERE clause.

Example:

```

SELECT a.empno, a.name, a.depno, b.depname
FROM employees a, department b
WHERE a.depno = b.depno

```

Outer Joins

The result set from an Outer join includes all the records in which the linked field value in both tables is an exact match. It also includes a row for every record in the primary table for which the linked field value has no match in the lookup table. For instance, you can use a Left Outer join to view all customers and the orders they have placed, but you also get a row for every customer who has not placed any orders. These customers appear at the end of the list with blanks in the fields that would otherwise hold order information.

Insert

With the INSERT statement you can enter new rows with data into a table.

Syntax

1) - get values from another table:

```

    INSERT INTO <table name1> (<column1>, <column2>, ...)
    SELECT <column3>, <column4>,...
    FROM <table name2>

```

2) Insert values directly:

```

    INSERT INTO <table name> (<column1>, <column2>, ...)
    VALUES (<value1>, <value2>)

```

Examples

Copy data from one table to another:

```

    INSERT INTO testdep (depno, depname, responsible, floor, client)
    SELECT depno, depname, responsible, floor, client
    FROM department
    WHERE depno = 200

```

Insert values:

```

    INSERT INTO testdep (depno, depname, responsible, floor, client)
    VALUES (200, 'Administration', 2, 5, 'NO')

```

Update

The UPDATE command will change data in the table rows specified in a WHERE statement.

Syntax

```

    UPDATE <table name>
    SET <column name>= <value>, ...
    WHERE <condition>

```

Example

```

    UPDATE testdep
    SET depname = 'Administration',
    responsible = 3
    WHERE depno = 200

```

Delete

The DELETE command removes rows identified by a WHERE <condition> clause.

Note: If the WHERE statement is omitted, all rows will be deleted.

Syntax

```

    DELETE FROM <table name> [WHERE <condition>]

```

Example

```

    DELETE FROM testdep WHERE floor = 3

```

Merge into

The MERGE command is an official SQL standard (introduced in SQL: 2003 - the 5th version of the SQL standard), and it is also included into the Agresso SQL standard.

Syntax

The formal syntax is as follows:

```

    MERGE INTO <table> <alias> USING <table> <alias>
    ON <join condition>
    WHEN MATCHED THEN
    UPDATE <set clause>

```

```
WHEN NOT MATCHED THEN
INSERT (<column list>) VALUES (<values list>);
```

Reference

See Release notes for Agresso Platform 2.2 for more details and examples.

COPY STATEMENT

COPY options

Overview

The COPY statement in ASQL is used to copy data between an Agresso table and a text (ASCII) file. The following options are available:

- COPY IN - copies data from a line separated text file to an Agresso table. Data for the various table columns are either in fixed positions in the text file, or tab separated.
- COPY TO - copies data from an Agresso table to a line separated text file. The data copied to the text file can either be given fixed positions or be tab separated.
- COPY FROM - copies data from a line separated text file to an Agresso table. The data in the text file are found in defined positions, but these positions may vary throughout the text file, depending on the value of a defined key.

Not to be used inside a transaction

Please note the none of the COPY options described here can be called inside a transaction.

COPY IN

The COPY IN statement copies data from an ASCII file and into an Agresso table.

Syntax

The general syntax is as follows:

```
copy in [import] file = <file name>,
colsep = <column separator>, table = <table_name>,
<col_name>[ = [s]<number>], ...
```

Parameters and keywords

The various syntax elements are described below:

Element	Description
IMPORT	If the IMPORT keyword is used, it tells the COPY statement to look for the file in the directory pointed to by the AGRESSO_IMPORT environment variable. In this case, no path is needed.
<file_name>	The name of the file you want to import from. If no path is given, it will look either in the current folder or where AGRESSO_IMPORT points to.
COLSEP	Column separator keyword. Must be set to a valid <column separator> value (F or T).
<column separator>	Indicates how data in the text file shall be interpreted. Valid values are: <ul style="list-style-type: none"> • F - Fixed positions, meaning that the data for a column occupies a fixed number of positions. The start position is always the next position after the previous column, or 1 if it is the first column. When Fixed positions are used, <col_name> parameter must always be followed by the (text) length to copy. • T - Tab separated.

<table_name>	Name of table to copy data into.
<col_name>	Name of column to copy data into.
S	<p>Used for a specific column. The S switch can only be used for one column, and indicates that the column shall contain a sequence number, and not data from the text file.</p> <p>If the S is followed by a number, this number will be the first number in the sequence (for the first row). For subsequent rows, the sequence number will increase with 1.</p> <p>If the S switch is used, it must always be followed by the first sequence number.</p> <p>Note: The column to use for sequence numbers, must have been defined as an int column!</p>
<number>	<p>Used in conjunction with the COLSEP value F. When the file has a fixed format, the value for every column takes up the same number of bytes in every row. This number specifies the number of characters (text length).</p> <p>An integer value can have up to 18 digits, and a float or money value up to 25 digits, and they are both stored as character strings in the file.</p> <p>Remember: When the S switch is used, it does not refer to the length, but to the first sequence number.</p>

Example

In the following example, we copy data from `myDataFile.txt`, located in the current folder - or in the import folder. The data have fixed positions, and we copy data from position 1 to 33. The third column in the table will hold a sequence number, starting with 1 for the first row:

```
copy in file = 'myDataFile.txt', colset = F, table = hlptab,
    account=8,
    amount=25,
    sequence_no=s1
/
```

COPY TO

The COPY TO statement copies data from a table and into an ASCII file.

Syntax

The general syntax is as follows:

```
copy to [import] file = <file_name>,
    colsep = <column separator>,
    select <col_name>[ = <number>], ...
    from <table name>[, <table name>]
    [where <condition>]
```

Parameters and keywords

The various syntax elements are described below:

Element	Description
IMPORT	If the IMPORT keyword is used, it tells the COPY statement to create the file in the directory pointed to by the AGRESSO_IMPORT environment variable. In this case, no path is needed.
<file_name>	The name of the file you want to copy to. If no path is given, it will either be created in the current folder or in the folder pointed to by the AGRESSO_IMPORT environment variable.
COLSEP	Column separator keyword. Must be set to a valid <column separator>

	value (F or T).
<column separator>	Indicates how data in the text file shall be separated. Valid values are: <ul style="list-style-type: none"> • F - Fixed positions. • T - Tab separated.
SELECT	SELECT statement used to identify the table(s) and columns to copy from. Apart from special syntax related to the copy operation, this is a standard SELECT, allowing you to use any type of conditions to filter out unwanted rows.
<col_name>	Name of column to copy from. If the <column separator> is set to F , you must also specify the number of characters to write to the text file.
<number>	Used when the colsep value is F . This number specifies the number of characters (text length) to use for the column value in the text file. This means that shorter values will be padded with blanks.
<table name>	Name of table(s) to copy from.

Example

In the following example, we copy data from two columns in the table hlptab to the file <C:\AGRCOPY.txt> - but only rows where the value of the **amount** column is higher than **80000**.

```
copy to import file = 'c:\AGRCOPY.txt',
colsep = F,
select account=8,
        amount=25
from hlptab
where amount > 80000
/
```

COPY FROM

The COPY FROM statement copies data - on variable format - from a text file to a table.

Syntax

```
copy from [import] = <file_name>,
table = <table_name>,
key = <key_startpos> - <key_endpos>,
{<key value> = {<col_name> <type>(<pos>),...} ...}
```

Parameters and keywords

The various syntax elements are described below:

Element	Description
IMPORT	If used, it tells the COPY statement to create the file in the directory pointed to by the AGRESSO_IMPORT environment variable.
<file_name>	The name of the file you want to import from. If no path is given, COPY FROM will look either in the current folder or where AGRESSO_IMPORT points to.
<tab_name>	Specifies the name of the table you want the data to be copied into.
KEY	The KEY keyword is used to identify the start and end position of the row identifier (key!). Although other data fields may be located on different positions throughout the file, the key must always be available within the same start and end position.
<key_startpos>	Start position for the key value. The first character on a row is in position 1

<key_endpos>	End position for the key value.																										
<key value>	A specific key value which introduces a new interpretation of the data. When used, the <key value> must be followed by one or more column specifications. See <col_name>, <type> and <pos> below.																										
<col_name>	Name of column to copy data into. Must be followed by the column data type (tells how to interpret the text data) and the start and end position for the column data in the text file.																										
<type>	Specifies the data type of the column. Valid type values are: <table border="1"> <thead> <tr> <th>Type</th><th>Description</th></tr> </thead> <tbody> <tr><td>c</td><td>Character</td></tr> <tr><td>i</td><td>Integer</td></tr> <tr><td>f<no></td><td>Float with <no> decimals, no decimal separator</td></tr> <tr><td>f.<no></td><td>Float with <no> decimals and a decimal separator</td></tr> <tr><td>m<no></td><td>Money with <no> decimals, no decimal separator</td></tr> <tr><td>m.<no></td><td>Money with <no> decimals and a decimal separator</td></tr> <tr><td>d1</td><td>Date on the format MMDD.</td></tr> <tr><td>d2</td><td>Date on the format YYMMDD.</td></tr> <tr><td>d3</td><td>Date on the format YYYYMMDD.</td></tr> <tr><td>d</td><td>Date on the format YYYYMMDD hh:mm:ss.</td></tr> <tr><td>p1</td><td>Period on the format YYPP.</td></tr> <tr><td>p2</td><td>Period on the format YYYYPP.</td></tr> </tbody> </table>	Type	Description	c	Character	i	Integer	f<no>	Float with <no> decimals, no decimal separator	f.<no>	Float with <no> decimals and a decimal separator	m<no>	Money with <no> decimals, no decimal separator	m.<no>	Money with <no> decimals and a decimal separator	d1	Date on the format MMDD.	d2	Date on the format YYMMDD.	d3	Date on the format YYYYMMDD.	d	Date on the format YYYYMMDD hh:mm:ss.	p1	Period on the format YYPP.	p2	Period on the format YYYYPP.
Type	Description																										
c	Character																										
i	Integer																										
f<no>	Float with <no> decimals, no decimal separator																										
f.<no>	Float with <no> decimals and a decimal separator																										
m<no>	Money with <no> decimals, no decimal separator																										
m.<no>	Money with <no> decimals and a decimal separator																										
d1	Date on the format MMDD.																										
d2	Date on the format YYMMDD.																										
d3	Date on the format YYYYMMDD.																										
d	Date on the format YYYYMMDD hh:mm:ss.																										
p1	Period on the format YYPP.																										
p2	Period on the format YYYYPP.																										
<pos>	The position is in the <start_pos> - <end_pos> form, for example 12-23. The first position in a row is 1. Remember that mycol (10-15) means that the mycol column in the table have length 6, starting in position ten, including position 15.																										

Example

In the following example we copy from the file Tbrk0606.dat located in the current folder or in the import folder. The KEY is located in first 8 positions, and we can handle three key values (meaning that other key values will be ignored):

```
COPY FROM IMPORT = 'Tbrk0606.dat',
  TABLE = 'hlptbl',
  KEY = 1-8,
  '940SWI01' = foreign_acc c(42-76),
    statement_ob c(110-134)
  '940SWI02' = foreign_acc c(42-76),
    trans_date d2(91-96),
    voucher_date d1(97-100),
    dc_cflag c(101-101),
    cur_amount m2(104-118),
    description c(141-156),
    ext_inv_ref c(157-162)
  '940SWI05' = foreign_acc c(42-76),
    statement_cb c(91-115)
```

! Note that there should be no comma between each new key value.

AGRINDEX

Re-generation of indexes

General purpose

AgrIndex.exe is a console application for re-generation of indexes for Agresso tables. When run, AgrIndex will collect the index definitions from the tables `asysindex` and `aagindex` and verify that all indexes in the database exist according to the definitions.

If a defined index is missing, AgrIndex will re-create it. If the index exist, but the index column definition is wrong, the column will be re-created correctly.

Index definition tables

Agresso index definitions are found in the table `asysindex`, which contains the indexes distributed by the Agresso installation package. User defined index definitions should be placed in the `aagindex` table.

Application location

A standard ABW installation will place *AgrIndex.exe* in the bin directory (AGRESSO_EXE).

Using AgrIndex

Basic syntax

The syntax for running AgrIndex is as follows:

```
Agrindex.exe <DataSource> [-s | -a | -u] [-t<location> | -T<location>] [-n]
[<table filter>]
```

where `<DataSource>` and at least one parameter (`-s` | `-a` | `-u`) are mandatory.

Parameters

The table below explains the parameters (or switches) you can use:

Parameter	Description
<code>-s</code>	Indexes found in <code>asysindex</code> are regenerated
<code>-a</code>	Indexes found in <code>aagindex</code> are regenerated
<code>-u</code>	Indexes found in both <code>asysindex</code> and <code>aagindex</code> are regenerated (union)
<code>-t<location></code>	Specify database location. This will be used for locations not found on existing indexes or if the index is missing.
<code>-T<location></code>	Specify a database location that will be used for all re-generated indexes.
<code>-n</code>	Don't use <code>acrclient</code> location for system tables (<code>acrclient</code> 's location is used for system indexes by default).
<code><table filter></code>	A table filter, in the form of a table name (wildcards allowed) can be added to the end of the parameter list. Example: <code>acr%</code> limits the selection to tables starting with acr.

Examples of use

The following examples shows AgrIndex in use:

- To regenerate all Agresso indexes for data source MyServerDataSource:
`Agrindex.exe MyServerDataSource -s`
- To regenerate indexes for all tables starting with `aim` with location MY_LOCATION:
`Agrindex.exe MyServerDataSource -u -TMY_LOCATION aim%`

- To regenerate all user defined indexes (indexes defined in [aagindex](#)):
`Agrindex.exe MyServerDataSource -a`
- To regenerate indexs for the table [aagmenu](#)
`Agrindex.exe MyServerDataSource -s aagmenu`

Related topics

[Agresso Data Segments](#)

Visual Basic for Application (VBA)

VBA OVERVIEW

The VBA Installation package consists of

- VBA Microsoft Merge Module delivered from Summit Software and containing the VBA specific files (VBA version 6.4)
- Agresso VBA related files [AgrPrivate.dll](#) and [AgrPrivate2.ocx](#)

For VBA in a Centrally Configured environment or VBA in a Citrix server and Meta Frame environment please see relevant sections.

Limitation

VBA does not support the [BigInt](#) datatype or other 64-bits integer values.

SIGNING AND SECURITY

Warnings during load

When Agresso tries to load the VBA projects, you will often get a standard Microsoft warning, stating that the projects contain macros, and that they can be dangerous. If you are unlucky, you will get one warning per project.

This happens for all unsigned projects, or more exact, for all projects that are not signed by a trustworthy source.

You can handle this in several ways:

- Keep it as it is. You will have full control of the loading process, although the warnings may be a bit tiresome.
- You can set the security level for your VBA projects to Low. Agresso 5.5 introduced a new system parameter, [VBA_SIGN](#) (discussed later in this section), that allows you to do this, and thereby ensure loading without warnings.
- You can make sure that the projects are digitally signed by a trustworthy publisher. Thereby you not only avoid problems during load, but you will also get a better control of (unofficial) project updates.

VBA version control

The new system parameter [VBA_REQV](#) can now be used to require that a certain VBA version, or a newer version, is available on the client computer. If this parameter is activated and a version is set, older VBA versions will not load, and the VBA projects can not be run.

Extended load options

The system parameter [VBA_LEVEL](#) is used to set the load level for the available VBA projects. [VBA_LEVEL](#) has now got two more levels, supporting project signatures and enhanced control.

Microsoft VBA 6.4 and MSI

Agresso uses Microsoft VBA 6.4, which is not compatible with the old MSI files. You must remove the old MSI versions and Agresso Client before you can install VBA 6.4 on your client computer.

System Parameters

- **VBA_SIGN**. This parameter is used to set the acceptable security level for all available VBA projects.

Valid values are:

Value	Description
VH	Very High. Only macros installed in trusted locations will be allowed to run. All other signed and unsigned macros are disabled.
H	High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.
M	Medium. You can choose whether or not to run potentially unsafe macros.
L	Low. You are not protected from potentially unsafe macros. Use this only if you have checked the safety of all VBA projects.

Agresso comes with the following setup for VBA_SIGN:

Value	Length	Status
4		Deactivated

- **VBA_REQV** (required version). This parameter can be used to define a specific VBA version. Only versions like, or newer than, the defined one, will be allowed to run.

! *VBA_REQV will only have an effect if the system parameter VBA_LEVEL is activated with value 5 (see next parameter).*

Valid values are any valid VBA version.

Agresso comes with the following setup for VBA_REQV:

Value	Length	Status
25		Deactivated

- **VBA_VALUE**. This parameter is used to set the load options for VBA projects and has now been extended with two more values, 6 and 7. All values are explained below.

Valid values are:

Value	Description
0	Try to load Agresso VBA and all projects. If VBA is not installed on the client, a message box is displayed.
1	Do not load Agresso VBA.
2	Requires that the file agrCT32.dll is available in the Agresso executable directory. If not, the Agresso Office client will not start.
3	Requires that the VBA environment on the client is loaded successfully.
4	Try to load VBA and all projects. Do nothing if VBA is not available.
5	Check that the client's VBA is compatible with VBA_REQV (see previous parameter). If not, abort the load process.
6	Require that VBA and all projects, with another status than D=Design, are loaded successfully. If any errors occur, shut down Agresso. If errors occur, you will not be able to start the Agresso client and

	reset this value! Must be handled with care!
7	Require that VBA and all projects are loaded successfully, and that all projects are signed by a trustworthy source. If any errors occur, shut down Agresso. <i>If errors occur, you will not be able to start the Agresso client and reset this value! Must be handled with care!</i>

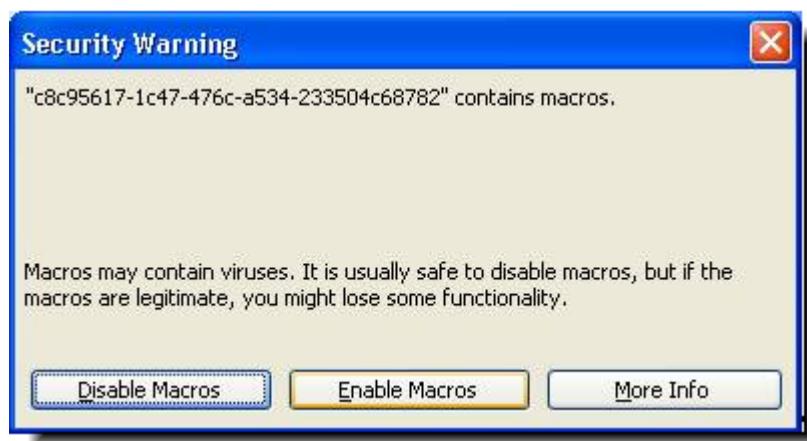
Signing of localisation specific VBA Projects in 5.5

In order to keep ABW at a high security level, Agresso files are signed with an R&D certificate. From 5.5 this also includes the localisation specific VBA projects delivered on the 551 DVD.

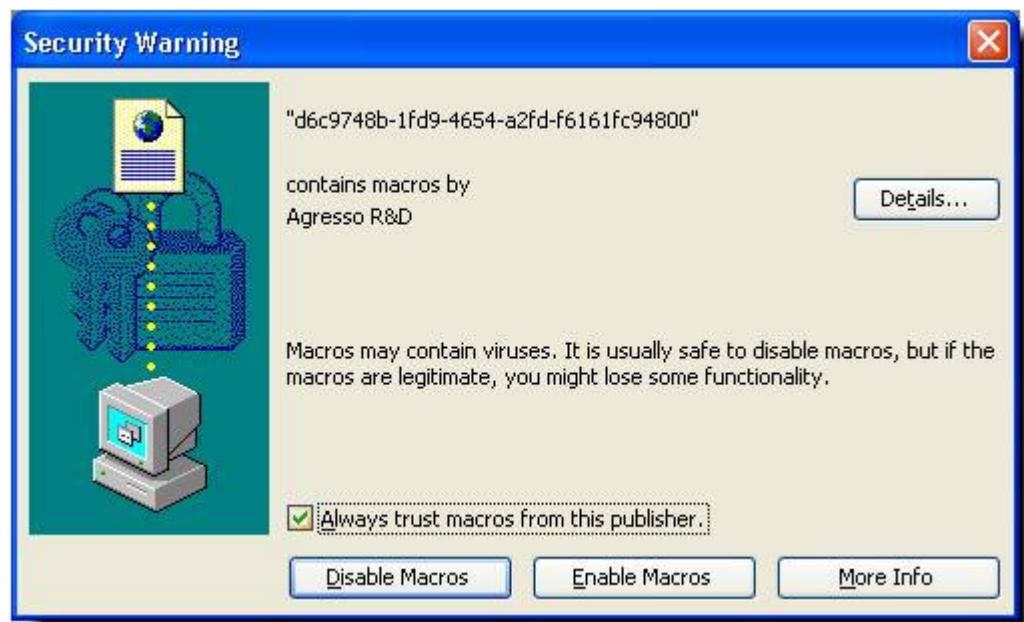
VBA projects delivered to customers as ordinary customisations, will *not* be signed by the R&D certificate.

Security warnings

A number of security warnings will appear when logging on to the Agresso client if non-signed VBA projects are active. The user will have to enable the use of each of these VBA projects on login as displayed below:



When the VBA projects are signed by R&D, the user will get the following message on first time logon to Agresso:



If the user then checks the box **Always trust macros from this publisher.**, the user has trusted VBA projects by Agresso R&D once and for all, and will have no more security warnings regarding the localisation specific VBA projects.

VBA PROJECT MAINTENANCE AND IMPORT EXPORT TOOLS

VBA Projects in Agresso

The VBA projects in Agresso Office are stored in the blob table *actblob*, where each project has a unique blob id.

There are several projects that are included in the Agresso database. Some of them are sample projects and some of them are related to the localisation. All new customer related projects will also be stored in the same table.

For questions regarding the VBA projects stored in the database, please contact your local Agresso support.

Export projects

Visual Basic Project Maintenance

	Name	Description	Project ID	VBA project	User	Updated	S
1	AgriVBAProjectImportExport	VBA project import and export utility	1	1	sysen	09/07/2001	P
2	UK_Payroll	UK Payroll	5	1	sysen	10/25/2001	P
3	UK_PaySlip	UK Pay Slip	6	2	sysen	10/25/2001	P
4	Formatter	Word Mail Merge	3	1	sysen	03/30/2001	P
5	HS01	Norwegian Payroll	4	1	sysno	06/15/2001	P
6	UK_SMP	UK Statutory Maternity Pay	2	2	sysen	10/25/2001	P
7	ShowVouch	Show Voucher Image	7	1	sysen	10/29/2001	N
8	ShowWFMap	Show Workflow Map	8	1	sysen	10/29/2001	N

In order to be able to export VBA projects directly from the Agresso client you must open Visual Basic Project Maintenance, mark the row(s) that you want to export and then select Export VBA Project on the Tools menu.

Export VBA Projects

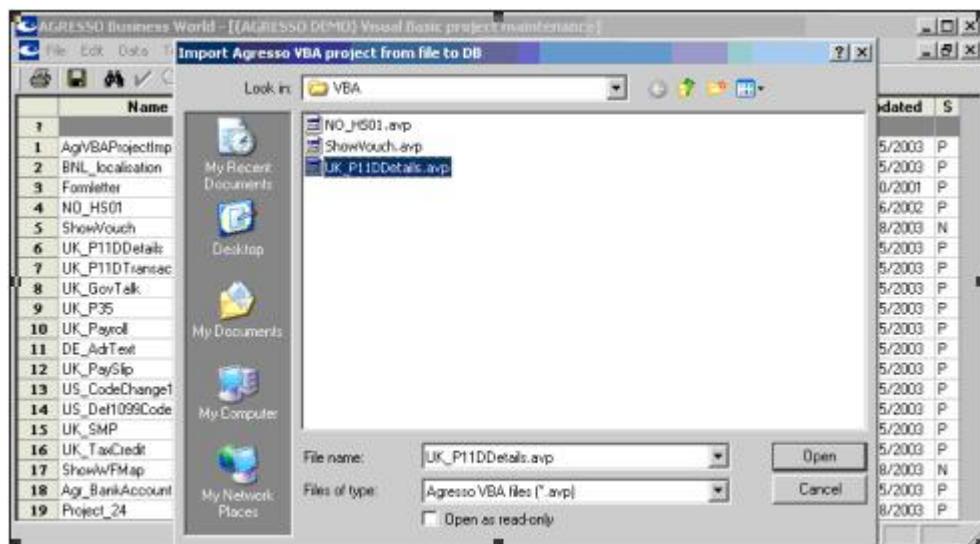
Name	Description	Project ID	VBA project type	User	Updated	S
1 AgriVBAProjectImportExport	VBA project import and export utility	1	1	sysen	01/15/2003	P
2 BNL_localisator	Cleanup Project type zero	17	1	sysen	01/15/2003	P
3 Formatter	Word Mail Merge	3	1	sysen	03/30/2001	P
4 NO_HS01	Norwegian Payroll	4	1	sysno	05/16/2002	P
5 ShowVouch	Show Voucher	7	1	sysen	01/28/2003	N
6 UK_P11DDetails	UK P11D Employee information	10	2	sysen	01/15/2003	P
7 UK_P11DTransactions	UK P11D Transactions	11	2	sysen	01/15/2003	P
8 UK_GovTalk	UK GovTalk	20	2	sysen	01/15/2003	P
9 UK_P35	UK P35 Information	21	2	sysen	01/15/2003	P
10 UK_Payroll	UK Payroll	5	1	sysen	01/15/2003	P
11 DE_AdrtText	German extra address fields	19	1	sysen	01/15/2003	P
12 UK_PaySlip	UK Pay Slip	6	2	sysen	01/15/2003	P
13 US_CodeChange1099	US Code Change 1099	15	1	sysen	01/15/2003	P
14 US_Def1099Code	US Def 1099 Code	16	1	sysen	01/15/2003	P
15 UK_SMP	UK Statutory Maternity Pay	2	2	sysen	01/15/2003	P
16 UK_TaxCredit	UK Tax Credits	9	2	sysen	01/15/2003	P
17 ShowWFMap	ShowWFMap (54sp2)	23	1	sysen	01/28/2003	N
18 Agri_BankAccount	overall country_code dependent bank_account check	22	1	sysen	01/15/2003	P
19 Project_24	c	24	2	sysen	01/28/2003	P

The selected files will be saved as .avp and .ini files in the folder you choose. If you save one file at a time you can change the file name. When you select more than one project the filenames will be the same as the selected projects.

Import projects

Open Visual Basic project maintenance and go to a valid row in the table. Select Import VBA Project on the Tools menu and select one or more projects from Import Agresso VBA project from file to DB.

Import Agresso VBA Project from File to DB



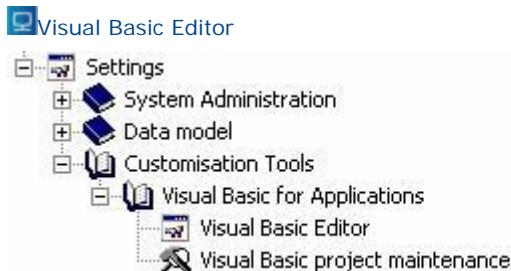
You should be able to update an already existing project with the same blob ID or project name. However you will not be able to save an already existing project with the same name but with another blob ID. You could manually change the blob_id in the .ini file, but keep the same format.

The VBA Editor

It is also possible to Import and export VBA projects from the VBA Editor.

In VBA for Agresso Office there is a totally integrated Import and Export functionality for VBA projects. You have access to import and export your projects directly from the VBA Editor. There are several advantages with this functionality. The projects can be imported directly with your special preferred blob ID. And this Integrated Import Export utility for VBA projects does not require any additional objects. That means that this functionality works on all operating system that Agresso Office supports.

The file format is as before one .avp file (Agresso VBA Project file) and one .ini file. The new functions are accessible in the VBA Editor from the file menu. Open the Visual Basic Editor from the Agresso Office menu tree.

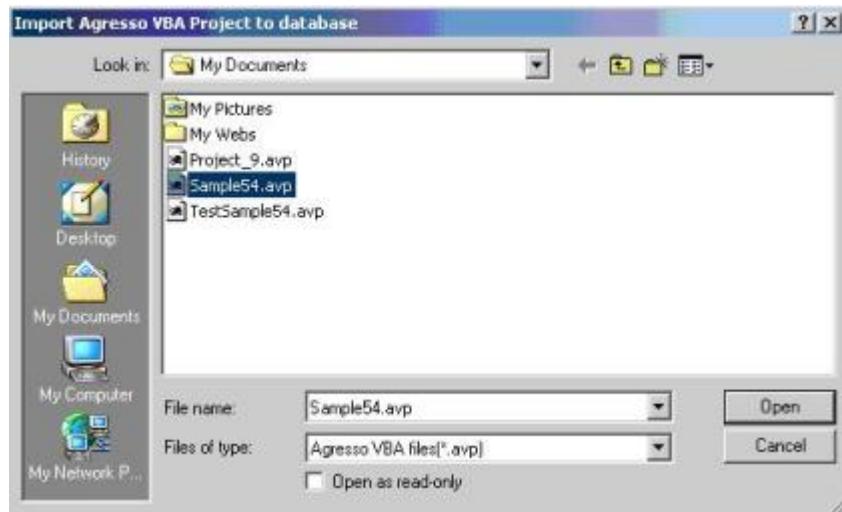


There are three menus in the File menu in the VBA Editor, Import Project to Database, Export Active Project from Database and Cleanup Project type zero.

IMPORT PROJECT TO DATABASE

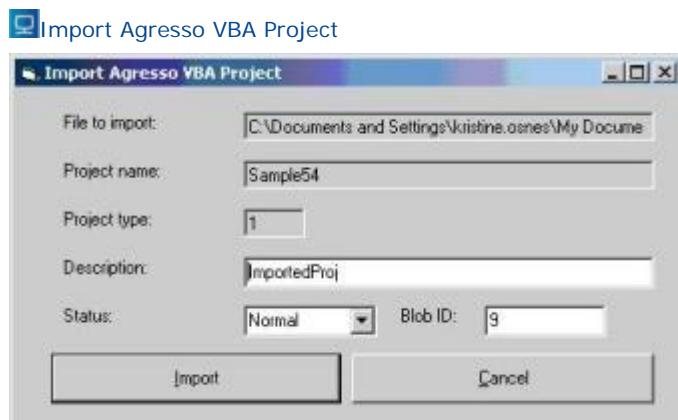
Make sure that both the .avp and .ini files for the project are located in the same directory before you try to import the project. Select the Import Project to Database menu item. An Open File dialog box will appear. Browse for the project you would like to import and press Open.





A new Dialog box is opened containing more information about the VBA Project. You have the following options:

- The File to import field is the path to the file that will be imported. This is a read only field.
- The Project name field is the name of the project. This is a read only field.
- The Project type is 1 for Application item project and 2 for Custom Form project
- In the Description field you can add a description for the project. It is not possible to add a description for Custom Form Projects.
- The status can be changed to Parked or Design in the Status field.
- In the Blob ID field you can define the project ID (or blob ID) for the project. This can be useful when the projects are connected to a special menu item in asysfunctions. The default blob ID is the next available number in the database. It is not possible to overwrite any existing project in the database



The VBA Application Item projects will not be loaded into the Agresso Office environment before Agresso is restarted. That is because the projects are loaded immediately after Agresso is executed.

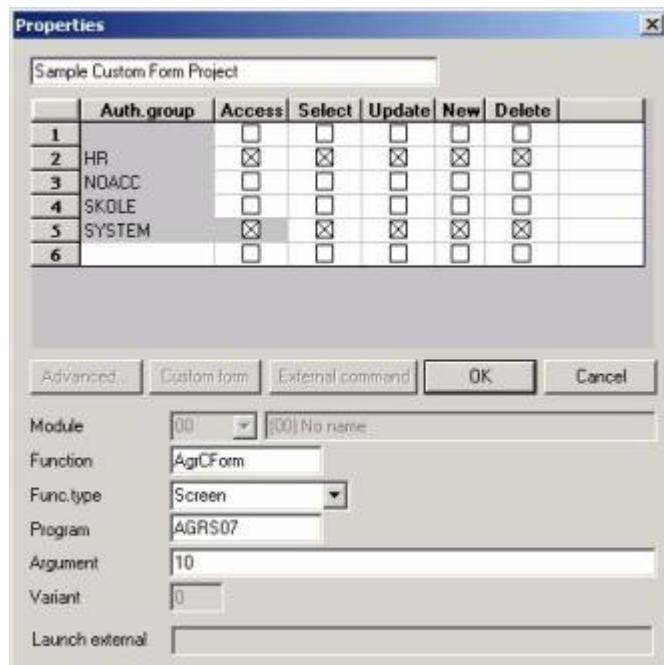
For Custom Form Projects (new windows developed using VBA in Agresso Office), you must also add a menu item manually to be able to access the window.

Add a Menu Item



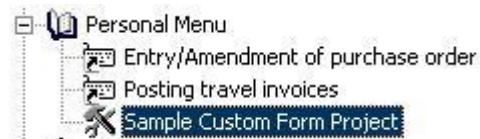
The new menu item is added by right clicking on the menu where you would like the Custom Form window to be stored. Select New menu. The Properties dialog box will appear.

Add Menu Item - Properties



Enter the name for the new menu item. Click the Custom form button. Add the correct ID in the Argument field. Press OK. The menu item will now be available from the Agresso Office menu tree.

Agresso Office Menu Tree



EXPORT ACTIVE PROJECT FROM DATABASE

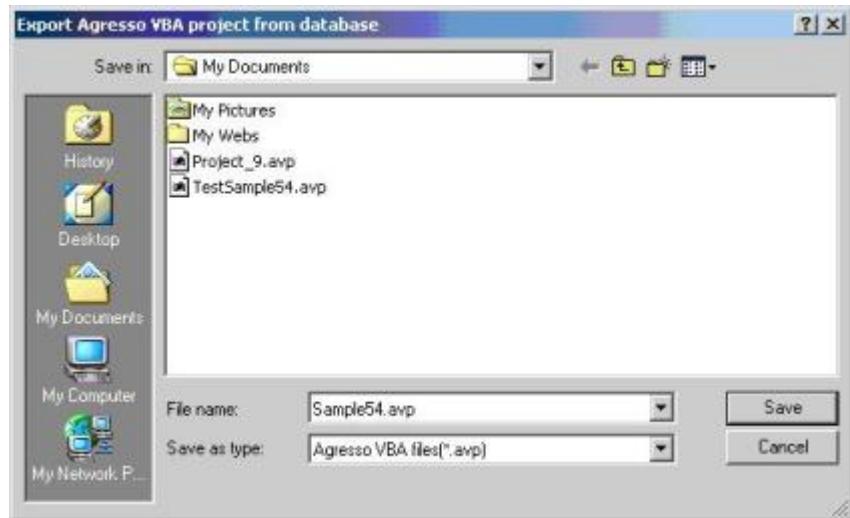
Choose the project you would like to export from the VBA Editor.

Choose Project to be Exported



Select the Export Active Project from Database menu item from the File menu in the VBA Editor. The Export File dialog box is opened. Browse for the location where you would like to store the project files. Press Save. The project is now exported.

Export Agresso VBA Project from Database



Security

SECURITY AREAS

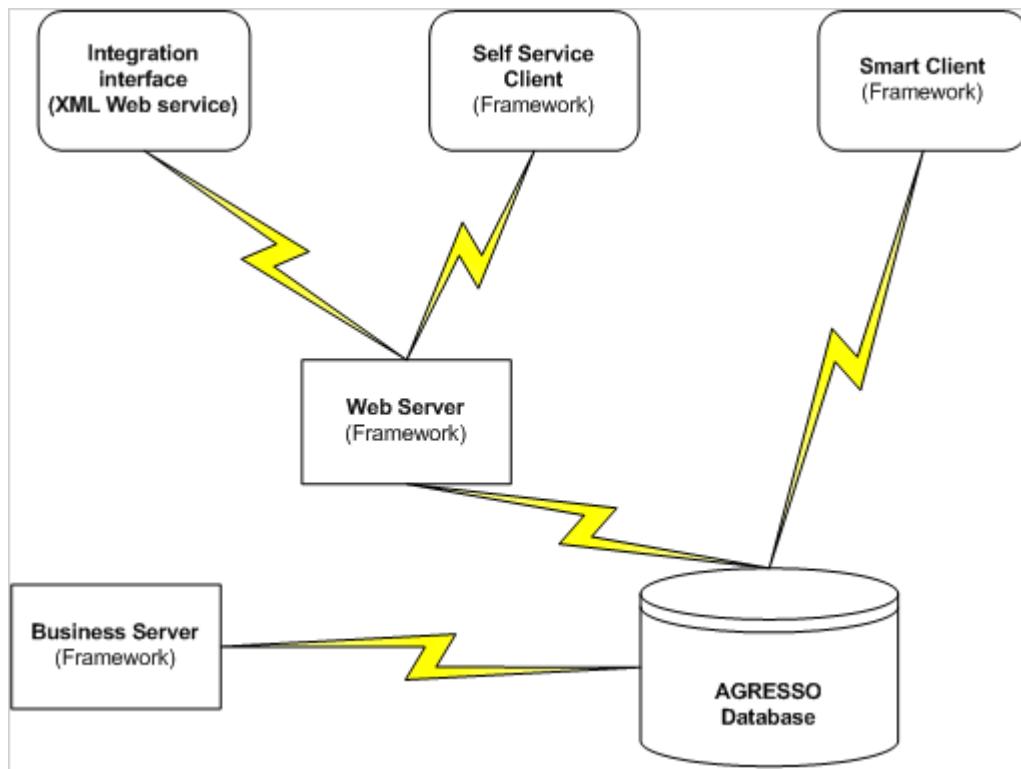
Basic architecture

From a security point of view, we can identify the following basic elements in a typical Agresso configuration:

- A set of client types:
 - Integration interface (a customised browser client)
 - Agresso Self service client
 - Agresso Smart client
- The Web server,
- The Business server,
- The Agresso database.

Communication diagram

The communication between the various elements are illustrated below:



Customer responsibility

Before implementing Agresso in a production environment, it is important that the installation owner has performed a thorough risk analysis, and has a high degree of security awareness.

Security areas

Agresso explicitly addresses two main security areas:

- Access control
- Program or module authentication.

These areas are managed by standard Agresso functionality and the Agresso Framework components, which makes sure that neither users nor program modules get access to the Agresso database unless properly authenticated.

In order to set up a secure Agresso system, it is also necessary to protect communication between the various elements. Agresso has no built-in functionality for protection of communication through insecure lines, but supports all the generally available tools for hiding and protecting communication over the network(s).

General recommendations

The following general rules should be applied:

- Run the system with the lowest privileges possible.
It is important to configure the system to run with the lowest privileges possible to reduce the damage if someone executes non-trusted code within the Agresso context.
- Configure your operating system as recommended by the vendor.
Please follow the recommended security guidelines from the operating system or dependent system vendors. Always install the latest patches and security updates.
- Assure that you have tightened your NTFS permissions and configure the code groups and permission sets at the most detailed level possible. Configure software restriction policies on your domain and assure that you have an appropriate physical protection for the system.
- Change the Agresso schema owner password at regular intervals.

- Use Centrally Configured Clients for Smart Client users; avoid stand alone client installations! For high security environments, we recommend Citrix / Terminal Server, with Smart Client as Published application.
- Use traffic encryption between clients and the application and database server.

Agresso Security

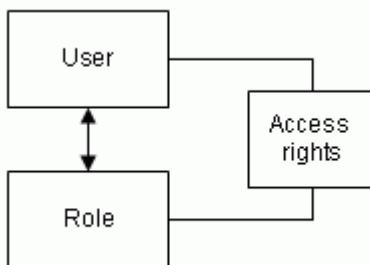
ACCESS CONTROL

Password protection

Users and roles

The basic access rights in Agresso are based on registered users and defined roles. A user may inhabit, or belong to, many roles, and a role will normally encompass many users. The access rights for a specific user are made up of the sum of access rights for the user as such, and for the roles the user belongs to. Both users and roles have a defined life-span, identified by a from-date and a to-date.

The relations between users, roles and access rights are outlined in the following model:



User authentication models

Agresso supports authentication models for both Active Directory (AD) and LDAP users, and has default solutions for log-on through specific Agresso log-on windows, as well as for single sign-on (SSO).

Password protection

When a person tries to connect to Agresso, Agresso will always check against registered user names and passwords.

Passwords are encrypted according to the company's selected encryption algorithm. Agresso is delivered with five predefined encryption variants (numbered 0 to 4, where the highest number points to the most secure algorithm). The company is free to introduce new ones, which will then be added to the list.

The PWD_VARIANT system parameter

The system parameter PWD_VARIANT holds a value that points to the companies default encryption algorithm. Upon delivery of Agresso, PWD_VARIANT will have the default value 4, pointing to the SHA-1 and salt encryption.

When new users are created, their password will be encrypted according to the value of PWD_VARIANT.

A note on ABW 5.4: In Agresso 5.4, the default variant value was 3. The new default variant value will not be used until the user changes his/her password.

Other parameters: See overview of password related system parameters [below](#).

Individual adjustments

The encrypted password is stored together with a pointer to the encryption variant used. An administrator can change the encryption variant for individual users, and thus disregard the default value of PWD_VARIANT.

Default password encryption

Salt and SHA-1

When using the (default) PWD_VARIANT 4, the user's password is *salted* and encrypted according to the SHA-1 algorithm before it is stored in the database.

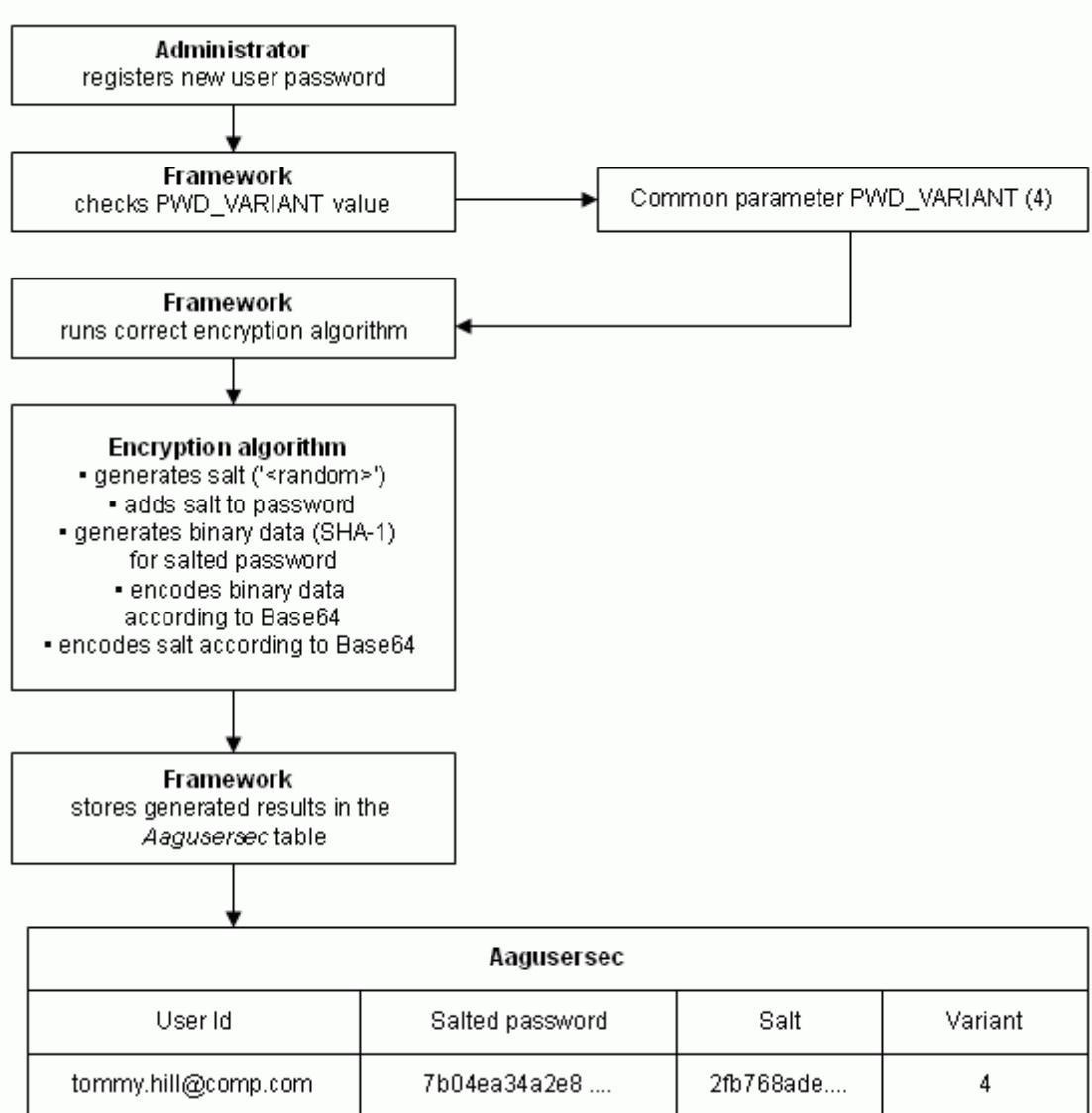
Encryption and storage

Salt is a random length string with random characters, generated when a new user is created or when the user changes the password.

This salt is added to the password – before encryption – and stored with the salted password and the PWD_VARIANT in the table Aagusersec.

Encryption example

The diagram below illustrates how a (new) password is encoded and stored according to the default value of PWD_VARIANT:



Management of roles and users

Reference

Management of users and roles in Agresso are handled by the User master file and Role master file respectively. In addition, we can use the Data control management master file to further detail access rights for roles and users.

Please refer to the Agresso Data control and user access reference manual for details. In this document we will just give a general overview of user and role management, in order to present a complete picture of security related issues.

ACCESS model

A user – or a role – is given access to a set of menu items, and if desired, to all underlying menu items in the menu tree. If necessary, an administrator can restrict access to specific menu items on lower levels.

Data control

Although a user (or role) may have full access to a certain master file, data control allows the administrator to restrict access to specific instances of the data, based on the data value.

Attribute based control: Data control is based on the functionality for attributes, attribute values and relations, and is implemented for all master files.

Password settings by system parameters

Parameter name	Description
PWD_CODE	If turned on, the password must contain both characters and digits.
PWD_DURATION	Sets the valid period – in days – for a new password.
PWD_EXPIRATION_WARNING	States – in number of days before the password expiration date – when a user shall be notified.
PWD_GENERATION	Sets the number of historic passwords that will be remembered when a new password is entered and validated. The new password cannot be any of the stored, historic passwords.
PWD_LOCKOUT_DURATION	Specifies - in minutes – for how long you will be locked out from Agresso after a defined number of invalid logon attempts. The number of invalid attempts is defined in the parameter PWD_TRIES.
PWD_LOG	Specifies whether a record of the accesses to Agresso is maintained. If the function is used the following will be logged: 1. Logons. 2. Normal logging on by use of Agresso's Exit function. 3. Lockouts. The report System access log (AG11) shows logged accesses and exits.
PWD_NOCHARS	Sets the minimum number of characters required for a valid password. If set to 0, there will be no check.
PWD_NOT_USERNAME	If turned on, the password cannot be the same as the user name.
PWD_TRIES	Sets the number of accepted, invalid logon attempts before the user is locked out. The same limitation is valid for all users.
PWD_VARIANT	A pointer to the security architecture where the encryption algorithm or hash is stored. When changing this variant, all new passwords (for new users and when users change their password) will be affected.

Supported Platform - Agresso Business World Route 66

For the latest update, please read the platform support list maintained by Agresso Product Services.

[Platform support list.](#)

AUTHENTICATION

Authenticators controls how users authenticate when they logon to Agresso. By default users login with Agresso user and password (Agresso Authentication) .

Another options is Windows Authentication (Referred to as Single-Sign-On in previous Agresso releases).

Configure Windows Authentication

To enable windows authentication, the Windows Authentication authenticator must be configured and moved to first priority.

Agresso authenticators are configured from the Smart Client. The menu item is located under Authentication in System administration.

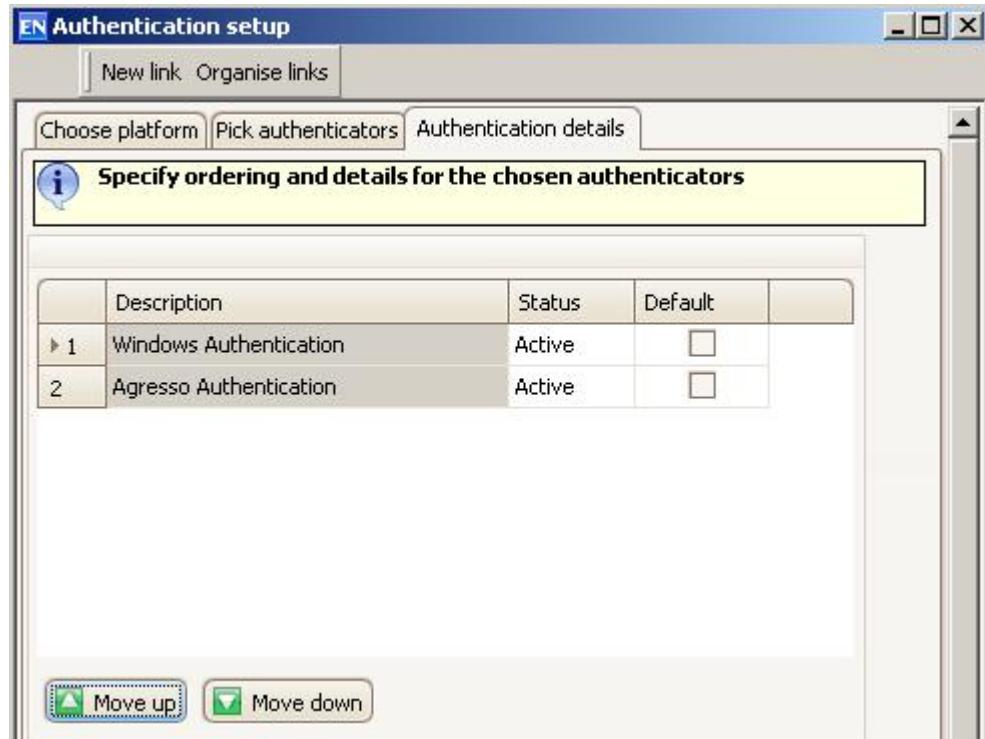


Existing authenticators are configured from the authentication setup page. The configuration is done in three steps:

1. Select the platform: Self Service, Smart Client or Web Services.
2. Select the authenticators that should be made available
3. Set the priority of the authenticators

To enable integrated windows authentication, select the windows authentication authenticator and move it to first priority.

Agresso users needs to be associated with a domain users to use windows authentication, this can be configured from **user master file** in the smart client.



Self Service and Web Services

For Self Service and web services an additional step is needed to enable or disable windows authentication.

Click the [Configure windows authentication](#) link (the **Authentication** tab) in the properties window for the web application

For details on the Authentication setup configuration, please refer to the Reference manual for authentication setup that can be found on the ABW 5.5.3 DVD.

Glossary

A

ADO: Abb. for ActiveX Data Objects. Microsoft's library for accessing data sources through OLE DB. Typically it is used to query or modify data stored in a relational database.

Agresso Management Console: Tool for managing Agresso installations; based on Microsoft Management Console

Agresso Parser: The process that translates Agresso SQL syntax into native database syntax

Agresso Server Datasource: A set of environment variables stored in registry; defines the environment in which the server processes are running

ASQL: Abb. for Agresso SQL

Automatic Undo Management Mode: A mode of the database in which undo data is stored in a dedicated undo tablespace. Unlike in manual undo management mode, the only undo management that you must perform is the creation of the undo tablespace. All other undo management is performed automatically.

B

Base Table: An autonomous, named table. A base table exists in its own right, unlike a table that is merely constructed as a result of a query (view)

C

Cartesian Product: The Cartesian Product of a set of n tables is the table consisting of all possible rows r such that R is the concatenation of a row from the first table, a row from the second table, ..., and a row from the nth table.

Cluster: A cluster generally comprises two or more computers, or "nodes."

Connect Descriptor: A specially formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

Connect Identifier: A name, net service name, or service name that resolves to a connect descriptor. Users initiate a connect request by passing a username and password along with a connect identifier in a connect string for the service to which they want to connect, for example: SQL> CONNECT username/password@connect_identifier

Connection Parameters: Database parameters that vary from platform to platform

D

Default Domain: The network domain within which most client requests take place. It can be the domain where the client resides, or a domain from which the client often requests network services. The default domain is also the client configuration parameter that determines what domain to append to unqualified network name requests. A name request is unqualified if it does not have a "." character within it.

Directory Information Tree (DIT): A hierarchical tree-like structure in a directory server of the Distinguished Names (DNs) of the entries.

Directory Naming: A naming method that specifies a directory server to resolve a net service name into a connect descriptor. The net service name is stored centrally in a directory server.

Directory Naming Context: A subtree that is of significance within a directory server. It is usually the top of some organizational subtree.

Directory Server: A Lightweight Directory Access Protocol (LDAP)-compliant directory server. A directory can provide centralized storage and retrieval of database network components, user and corporate policies preferences, user authentication, and security information, replacing client-side and server-side localized files.

E

Environment Variables: A set of values stored in registry and used by the server system to locate vital system files

External Procedures: A PL/SQL routine executing on a server can call an external procedure or function that is written in the C programming language and stored in a shared library. In order for the database to connect to external procedures, the server must be configured with a net service name and the listener must be configured with protocol address and service information.

G

Global Database Name: The full database name that uniquely distinguishes it from any other database in your network domain. For example: sales.us.acme.com where sales is the name you want to call your database and us.acme.com is the network domain in which the database is located.

I

Index: A table of pointers. Each row in the indexes point to a corresponding row in a data table

Input Parameters: Database parameters common to all platforms

Installation Type: An installation type is a predefined component set that automatically selects which components to install.

Interprocess Communication (IPC): A protocol used by client applications that resides on the same node as the listener to communicate with the database. IPC can provide a faster local connection than TCP/IP.

J

Join: A query in which data is retrieved from more than one table

L

Listener: A process that resides on the server and whose responsibility is to listen for incoming client connection requests and manage the traffic to the server. When a client requests a network session with a database server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

Local Naming: A naming method that resolves a net service name into a connect descriptor.

Logistics Invoice Processing Server: A server component configured to perform invoice processing

Logistics Stock Processing Server: A server component configured to perform stock processing

M

Manual Undo Management Mode: A mode of the database in which undo blocks are stored in user-managed rollback segments. In automatic undo management mode, undo blocks are stored in system-managed, dedicated undo tablespaces.

N

Naming Method: A resolution method used by a client application to resolve a connect identifier to a network address when attempting to connect to a database service.

Net Service Name: A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a username and password along with a net service name in a connect string for the service to which they want to connect: SQL> CONNECT username/ password@ net_service_name

O

ODBC: Abb. for Open DataBase Connectivity. A standard for accessing different database systems.

P

Parallel Queue: A queue that can run two or more reports simultaneously

Primary Index: An index on a primary key field

Protocol Address: An address that identifies the network address of a network object. When a connection is made, the client and the receiver of the request, are configured with identical protocol addresses. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address. It is important to install the same protocols for the client and the connection recipient, as well as configure the same addresses.

R

RAID: Abb. for Redundant Array of Independent Drives

RDBMS: Abb. for Relational Database Management System. A database based on the relational model developed by E.F. Codd. A relational database allows the definition of data structures, storage and retrieval operations and integrity constraints. In such a database the data and relations between them are organised in tables. A table is a collection of rows or records and each row in a table contains the same fields. Certain fields may be designated as keys, which means that searches for specific values of that field will use indexing to speed them up.

Report Queue: A queue running reports when ordered from the client

Report Server: A Report Server is a server component that controls all reporting processes (tasks)

S

SAN: A Storage Area Network (SAN) is a network designed to attach computer storage devices such as disk array controllers and tape libraries to servers. As of 2005, SANs are common in enterprise storage. There are two variations of SANs: 1. A network whose primary purpose is the transfer of data between computer systems and storage elements and among storage elements. A SAN consists of a communication infrastructure, which provides physical connections, and a management layer, which organizes the connections, storage elements, and computer systems so that data transfer is secure and robust. The term SAN is usually (but not necessarily) identified with block I/O services rather than file access services. 2. A storage system consisting of storage elements, storage devices, computer systems, and/or appliances, plus all control software, communicating over a network.

Scheduler Server: A server component checking for scheduled events and order the associated reports

Secondary Index: An index on any other field than the primary key field

Serial Queue: A queue running only one report at a given time

SID: See System Identifier

SQL: Abb for Structured Query Language. A standard interactive and programming language for getting information from and updating a database

Syntax: The structure of strings in a language

System Identifier: The database system identifier that distinguishes the database from all other databases on your computer. The SID automatically defaults to the database name portion of the global database name (sales in the example sales.us.acme.com) until you reach eight characters or enter a period. You can accept or change the default value.

T

Table: In a relational system, a table consists of a row of column headings, together with zero or more rows of data values

Transaction Processing Server: Server component configured to perform transaction processing

U

UNC: See Universal Naming Convention

Universal Naming Convention: The Universal Naming Convention provides a means to access files on a network without mapping the network drive to a drive letter. UNC names are constructed in the following manner: \\computer name\\ share name\\ filename

V

View: A (named) table that does not have any existence in its own right, but is derived from one or more underlying base tables

Index

A

Aagqueryopt 267
aagserverqueue 92
Aagserviceuri 138
Aagusersec 342
Access Suit WEB Interface 286
Acrmessagequeuedet 159
Acrmessagequeuehead 159
Active Directory 342
Add License 208, 233
AGR_ALLOW_ERR 73
AGR_DEBUGGING 73
AGR_LANGUAGE 73
AGR_LOG_FORMAT 73
AGR_TEMPDB 11, 73
AGR_TMP_SAVE 73
AGR_USE_REAL_TABLE 73
AGR_USE_REAL_TABLES 9
agrblob 9
AgrConfigServer 289
agrdata 9
agrdoc 9
AGRESSO Alert Server 106
AGRESSO Blob Viewer 44
AGRESSO Central Configured Client 115
AGRESSO Client Configuration 286
AGRESSO Common Files 286
AGRESSO Database Copy Wizard 47, 52
AGRESSO Demo 5
AGRESSO Demo Database 5
AGRESSO EVENT SERVER 106
AGRESSO Installation 7
AGRESSO Management Console 44
Agresso OCR 4
AGRESSO Online Help 162
AGRESSO Report Engine 24
AGRESSO Self Service 134
AGRESSO Smart Client 286
AGRESSO Software Distribution Server 286
AGRESSO Software Distribution Service 124
AGRESSO SQL 44
AGRESSO VBA 333
AGRESSO Work Schedule Upgrade 257
AGRESSO_COM 73
AGRESSO_CUSTOM 73
AGRESSO_EXCELERATOR 73

AGRESSO_EXE 73
AGRESSO_EXPORT 73
AGRESSO_IMPORT 73
AGRESSO_LOG 73
AGRESSO_OCR 73
AGRESSO_PRINT 73
AGRESSO_REPORT 73
agresso_role 12
AGRESSO_SCRATCH 73
AGRESSO_STYLESHEET 73
agresso32.ini 289
AgrEventServer 106
agrindex 9
agrscratch 9
agrtemp 9
agrtempdb 9
Agrtid 47
ALERSERVER_URL 106
AMC 30, 44
Applications 18
ASQL 44
Asysqueryopt 267
AUTHENTICATION 111, 345
Authorization Upgrade 214

B

Blob Viewer 64
Bypass_ujvc 291

C

CCC 121
CCC Client Wizard 111
CCC Client32 115
Central Configured Client 111
Centrally Configured Client 286
certificate 134, 293
Citrix NFuse 286
Citrix Presentation Server 286
Citrix Published Applications 286
Citrix Secure Gateway 286
Compello 210, 235
Convert Compello 210, 235
Copy In Procedure 47
Copy Out 52
CopyMS 22, 44
CopyOra 22, 44
Crystal Report 272
CURSOR_SHARING 12

D	
Demo.....	5
E	
Easy Connect.....	12
EXPAND_PATH_WITH_CLIENT.....	2, 73
Export Active Project.....	339
Extended MAPI.....	151
EZCONNECT.....	12
F	
firewall.....	289
Framework	31
H	
Help Configuration.....	162
HINT	267
HTTP proxy.....	2
I	
Import Project.....	337
IMS	2
Installation Packages	128
Installation wizard	15
L	
Latin1_General_CI_AI	11
logging.....	27
M	
Mail	159
Mail Configuration.....	151
Mail Profile.....	153
Mail Queu	2
Main Upgrade Wizard	179
Microsoft Outlook	151
Microsoft Windows 2003 Terminal Server	286
mirroring.....	8
MMC	28
msiexec.....	27
N	
.NET Framework.....	286
Network Load Balancing	143
NLB	143
NLS_LENGTH_SEMANTICS.....	12
O	
Online Help.....	24, 25
optional services.....	106
ORACLE.....	12
P	
Platform 2.2.....	119
powershell	5
Prerequisites	7
Printer.....	97
Printer Definitions.....	97
Q	
priority	345
R	
RAID	8
READ_COMMITTED_SNAPSHOT	11
Report Engine	4, 24
REPORT_TIMEOUT	106
Responsibles Upgrade	217, 245, 253
Retire Service.....	138
Run ASQL Scripts.....	55
S	
SAN	8
SDS	121, 124, 286, 289
SDS functionality	121
segment	9
Self Service	134, 138
SetupABW	4
SID.....	12
Signing.....	333
Single-Sign-On.....	345
Smart Client.....	20, 286
Installing	286
running.....	286
Smart Client Installation	20
SMS	159
SMS Configuration	155
SMS Profile	155
SMTP	2, 151
SMTP Gateway	155
SMTP Mail	155
Software Distribution Server	122
SQL Server	11
Sqlnet.ora.....	12
SSL.....	2
Supported SMS	155
T	
tablespaces.....	9
tempdb	9
Terminal Server.....	286
Timesheet Upgrade	260
troubleshooting	5
Trusted Certificate Database	134
U	
Unicode	12
UPDATE	291
Upgrade wizard	199, 223
user master file	345

V	
VBA	107, 333
VBA Projects	336
VBA Runtime.....	18
VBA_SIGN	333
View licence.....	208, 233
Vista	5
Visual Basic For Application.....	333
W	
Web Service.....	138
Web Site	144
Web.config.....	31, 134, 138, 144
Windows 2003 Server	286
Windows 2003 SP1	286
Windows Authentication	345
WMI	28
WMI Service.....	124
Workflow	210, 235