Installing Software AG Products On Premises

Version 10.15

October 2022

This document applies to Software AG Installer 10.15 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: SAG-INST-1015-20221015

Table of Contents

About this Guide

This guide provides pre-installation, installation, and uninstallation instructions for the following:

- webMethods products. For BigMemory Max and Terracotta, this guide covers installation only when they are being used with other webMethods products.
- Zementis Predictive Analytics when being used with webMethods products.
- Apama and MashZone NextGen.

For instructions on installing BigMemory Max, Terracotta, and Zementis Predictive Analytics in other contexts, and for instructions on installing products not covered in this guide, see the documentation for those products.

This guide does not provide instructions on configuring products after installation. For those instructions, see the product documentation.

Important:
If you want to upgrade products, you must use the instructions in *Upgrading Software AG Products*. The instructions in this installation guide do not cover upgrades, so following the instructions in this installation guide for upgrades would have unpredictable results.

The statement below applies to the Database Component Configurator:

With respect to the EU General Data Protection Regulation (GDPR), our product does not store, collect or process any personal information.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format *folder.subfolder.service*, APIs, Java classes, methods, properties.
Italic	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
`Monospace font`	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.

Convention	Description
|	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the | symbol.
[ ]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [ ] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at https://documentation.softwareag.com.

In addition, you can also access the cloud product documentation via https://www.softwareag.cloud. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at https://knowledge.softwareag.com.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at https://techcommunity.softwareag.com. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at https://github.com/softwareag and https://hub.docker.com/u/softwareag and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at https://empower.softwareag.com. Many services on this portal require that you have an account. If you do not yet have one, you can request it at https://empower.softwareag.com/register. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 Using Command Central or Software AG Installer to Install Products

Using Command Central to Install Products

Use Command Central in distributed staging and production environments to install, patch, configure, manage, and upgrade Software AG products; to create database components; and to connect products to database components. With Command Central, you can perform these tasks on multiple remote machines from one location.

Using Software AG Installer to Install Products

Use Software AG Installer to install Software AG products locally on standalone development machines.

2 Hardware Requirements and Operating System, Browser, and RDBMS Support

Hardware Requirements

Minimum and Recommended Hardware Requirements

Important:
The hardware requirements in this section are for the listed products only. They do not include 2GB of hard drive space for Software AG infrastructure shared libraries that are required by most products and that are installed once per installation directory. They do not include requirements for supporting software such as RDBMSs. They do not include the additional requirements listed in “Additional Hardware Requirements” on page 14. You must determine the overall hardware requirements for each of your machines based on the overall requirements of your operating system and software.

The table below lists the minimum and recommended (in parentheses) hardware requirements for your products; products not listed have minimal or no hardware requirements (for example, products that are hosted on Integration Server).

Product	Hard Drive Space	RAM	Cores
ActiveTransfer Agent	10MB	512MB	1
ActiveTransfer Server/Gateway	100MB (200MB)	512MB	1
AgileApps*	21GB	8GB	1 (4)
Apama			
■ Server	500MB	4GB	2
■ IAF & Adapters	20MB	4GB	2
■ Dashboard Servers	50MB	4GB	2
■ Software AG Suite Integration	10MB	4GB	2
■ Cumulocity IoT Integration	30MB	4GB	2
■ Capital Markets Adapters (each)	5MB	4GB	2
■ Capital Markets Foundation	55MB	4GB	2
■ Dashboard Viewer	400MB	4GB	2
■ Platform Manager Plug-in	5MB	4GB	2
■ Predictive Analytics Plug-in	25MB	4GB	2
API Gateway**	2GB	1.75GB	1 (2)
API Portal	20GB	8GB	4
ApplinX			

Product	Hard Drive Space	RAM	Cores
■ Server	100MB	80MB+300KB per user	2 (4)
■ Administration	75MB	40MB	
■ JSP Framework	50MB	2GB	
■ C#.NET Framework	55MB	2GB	
■ VB.NET Framework	30MB	2GB	
■ Visual Studio Add-In	1MB	2GB	
BigMemory Max	200MB (6GB)	3GB (5GB)	2 (8)
CentraSite			
■ Registry Repository	3GB	4GB (8GB)	1 (2)
■ Application Server Tier	3GB	4GB (8GB)	1 (2)
Software AG Designer Eclipse platform	500MB (1.5GB)	1.5GB (2GB)	1 (2)
Developer Portal	20GB	2GB	1
EntireX			
■ Core Files	100MB	2GB	1
■ Application Monitoring Data Collector	5MB	250MB	1
■ Command Line Scripts	2MB	100MB	1
■ Mainframe Broker Monitoring	50MB	500MB	1
■ Broker	20MB Windows, 30MB UNIX	1GB	2
■ Development Tools	5MB Windows, 10MB UNIX	1GB	1
■ Mini Runtime	10MB	1GB	1
■ Web Services deployed in Software AG Runtime***	140MB Windows, 150MB UNIX	1GB	1
■ EntireX plug-in to Software AG Designer	50MB	50MB	
Integration Server	300MB (500MB)	1GB (2GB)	1 (2)
Locate	20GB (40GB)	512MB (4GB)	1 (3)
MashZone NextGen	2GB	2GB	2

Product	Hard Drive Space	RAM	Cores
Microservices Runtime	70MB (100MB)	1GB	1
Mobile Designer	1GB	1GB	1
My webMethods Server	1G (2G)	2GB (4GB)	2 (4)
webMethods OneData			
■ Server	512MB	2GB (4GB)	1 (2)
■ MDR Modules	50MB		
■ Customer MDM Template	20MB		
Optimize			
■ Analytic Engine	1GB	4GB (8GB)	2 (4)
■ Infrastructure Data Collector	300MB	2GB	1
■ Web Service Data Collector	100MB	256MB	1
Platform Manager	500MB	128MB	1
Tamino	5GB	1GB	1
Terracotta	500MB (10GB)	2GB (4GB)	2 (8)
Trading Networks Server	50MB		1
Universal Messaging realm server when used with webMethods products	500MB	2GB	2
Zementis Predictive Analytics	5GB	4GB	1

*Includes hard drive space for Integration Server, with which AgileApps is co-hosted.

**Includes 300MB hard drive space and 512MB RAM for the API Data Store, 130MB hard drive space for Microgateway, and 200MB hard drive space and 256MB RAM for Kibana.

***Includes 140MB hard drive space, 1GB RAM, and 1 core for the required and automatically installed Web Services Stack.

Additional Hardware Requirements

The table below lists hardware requirements beyond those listed in the previous section.

Product	Additional Hardware Requirements
ActiveTransfer Server	ActiveTransfer Server needs additional hard drive space if you use the hard drive as your virtual file system. The space required depends on the total

Product	Additional Hardware Requirements
	size of documents ActiveTransfer Server needs to store, upload, and download. ActiveTransfer Server might need additional hard drive space for log files.
AgileApps	See the AgileApps wiki.
Apama	Hard drive space, RAM, and core requirements for Apama vary significantly depending on the Apama application you run. You will need to determine processing and memory requirements by measuring the application, since these requirements vary. Contact Software AG Professional Services for help.
API Portal	API Portal uses Elasticsearch. You will need 10GB of hard drive space to prevent Elasticsearch from going into read-only mode.
BigMemory Max	Hard drive space, RAM, and core requirements for BigMemory Max vary significantly depending on the webMethods products and customized applications you are using BigMemory Max with. You will need to determine processing and memory requirements by measuring the applications, since these requirements vary. Contact Software AG Professional Services for help.
Software AG Designer	For Software AG Designer Eclipse platform, 700MB additional free hard drive space is required in the system temp directory(Windows) or temporary (UNIX) directory. Each product plug-ins you install in Software AG Designer can require from 50K to 150MB of additional free hard drive space.
Deployer	Hard drive space for the deployment projects you create. When allocating space, allow for the number of projects to increase over time.
Locate	The hard disk space for Locate must be in a RAID configuration. The range of hardware requirements in the table is intended to accommodate your needs up to the largest dataset you could install.
MashZone NextGen	The RAM requirement correlates with the amount of data processed and might need to be increased based on use case. You might need to increase cores based on the number of simultaneous users.
Mobile Designer	You will need additional hard drive space to accommodate the SDKs for the mobile devices for which you want to develop applications.
webMethods OneData	webMethods OneData needs additional hard drive space for temp, import, and export files in the *Software AG_directory* \OneData\config directory.
Optimize	
■ Analytic Engine	The Analytic Engine needs 1GB virtual swap space and a disk subsystem with 10GB available free space. In a production environment, the disk

Product	Additional Hardware Requirements
	subsystem must have a redundant array of independent disks (RAID 5), an UltraWide SCSI, and 10K to 15K RPM drives. Also in a production environment, the Analytic Engine needs additional hard drive space for log files. The recommended amount is 100MB; the engine needs 50MB of space for each log file.
■ Web Service Data Collector	Each Web Service Data Collector needs 128MB of virtual swap space. In a production environment, each Web Service Data Collector needs additional hard drive space for log files. The recommended amount is 75MB; each Web Service Data Collector needs 5MB of hard drive space for each log file. More hard drive space might be needed if you use debug level or higher logging.
Terracotta	Hard drive space, RAM, and core requirements for Terracotta vary significantly depending on the webMethods products and customized applications you are using Terracotta with. You will need to determine processing and memory requirements by measuring the applications, since these requirements vary. Contact Software AG Professional Services for help.
Trading Networks	To use large document handling, Trading Networks Server needs additional hard drive space on which to temporarily save documents (instead of storing them in memory). For detailed information, see *webMethods Trading Networks User's Guide*.
Universal Messaging	If you use persistent topics or queues, or persistent messages, Universal Messaging needs additional hard drive space to persist the published data. The amount of space required would be loosely based on this equation: messages per second x message size x message time to live. If you store messages in memory only, Universal Messaging needs additional heap within the JVM to hold references to these messages. The heap size required is based on the same equation. By default the Universal Messaging server is configured to use 1GB of heap, and maximum 1GB for direct (off-heap) memory, hence the requirement for 2GB RAM by default. 2 CPU cores is the minimal requirement for running a production Universal Messaging server. Based on the load on the system, you may need to increase the cores.
Zementis Predictive Analytics	Additional storage may be required for working and temporary space. The amount of the total required disk space depends on the size of files (models, data, etc.) expected to be uploaded to and processed by Zementis Predictive Analytics.

Operating System Support

For information on operating systems that are supported by your products, see *System Requirements for Software AG Products*.

For Apama, also see *Apama Supported Platforms*.

If the vendor for your operating system recommends installing the latest service packs, patch-bundles, and fixes, Software AG encourages you to follow that recommendation. If you experience problems with installed service packs, patch-bundles, or fixes, report those problems directly to the vendor.

Browser Support

For information on browsers that are supported by your products, see *System Requirements for Software AG Products*.

RDBMS Support

For information on RDBMSs that are supported by your products, see *System Requirements for Software AG Products*.

3 Preparing for Product Installation

Shut Down Products

Shut down all non-Software AG applications that are running on the machine on which you are going to install. If these applications are not shut down, product files might become corrupted.

If you are going to install products into an existing Software AG product installation directory (that contains products from the same release), shut down running products in that directory so files that are locked by the operating system can be updated. For instructions, see the product documentation.

Prepare for Installation For All Products (Windows System)

Install the latest version (14.26.28720.3 or later) of the Microsoft Visual C++ Redistributable for Visual Studio 2019.

> **Note:**
> If you have an older version installed and did not shut down all applications that use it, you will have to reboot your machine and restart the installation process, which will then install this package for you.

Prepare for Apama Installation (Linux System)

The file system to which the Apama correlator writes its log files and persistence database can significantly affect the performance of latency-critical applications. File systems such as ext3, ext4, NFS, ZFS and GFS2 can have poor worst-case latency, especially when the system is loaded. For latency-critical applications, Software AG recommends using XFS where possible.

Prepare for API Gateway and Developer Portal (Linux System)

API Gateway and Developer Portal come with a data store, called API Data Store, in which they store core data, events, and metrics. The data store uses Elasticsearch 8.2.3. Elasticsearch cannot be run as the root user on a Linux system, so you must create a data store user and install and run the data store as that user.

Elasticsearch does several checks before starting up. Software AG recommends that you review the bootstrap checks (search on "Elasticsearch bootstrap checks") and important system configuration settings (search on "Elasticsearch important system configuration") before starting the data store. In particular, you may need to adjust these settings:

- Check the settings for the system-wide maximum number of file descriptors (kernel parameter fs.file-max) by executing the command `sysctl -a | fgrep fs.file-max`. If the value is less than 65536, log on as the root user and increase the value by executing `sysctl -w fs.file-max=200000` or `echo "fs.file-max=65536" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.

- Check the data store user settings for the maximum number of open file descriptors by executing the commands `ulimit -Hn` and `ulimit -Sn`, where `-Hn` is the hard limit and `-Sn` is the soft limit. If the value is less than 65536, log on as the data store user and increase the value to at least

65536 by executing `ulimit -n 65536`. To permanently save this setting for the user, execute the following:

```
echo "user_name soft nofile 65536" >> /etc/security/limits.conf
echo "user_name hard nofile 65536" >> /etc/security/limits.conf
```

- Check the setting for the system-wide maximum map count (kernel parametervm.max_map_count) by executing the command `sysctl -a | fgrep vm.max_map_count`. If the value is less than 262144, log on as the rootuser and increase the value to at least 262144 by executing `sysctl -w vm.max_map_count=262144` or `echo " vm.max_map_count=262144" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.

- Check the data store user settings for the maximum number of processes by executing the command `ulimit -u`. If the value is less than 4096, log on as the data store user and increase the value to at least 4096 by executing `ulimit -n 4096`. To permanently save this setting for the user, execute the following:

  ```
  echo "user_name soft nproc 4096" >> /etc/security/limits.conf
  echo "user_name hard nproc 4096" >> /etc/security/limits.conf
  ```

Prepare for API Portal Installation (UNIX System)

- Check the setting for shared memory (kernel parameter shmmax) by executing the command `sysctl -a | fgrep kernel.shmmax`. If the value is less than 629145600, log on as root user and increase the value by executing `sysctl -w kernel.shmmax=629145600` or `echo "kernel.shmmax=629145600" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.

- Check the settings for the system-wide maximum number of file descriptors (kernel parameter fs.file-max) by executing the command `sysctl -a | fgrep fs.file-max`. If the value is less than 200000, log on as the root user and increase the value by executing `sysctl -w fs.file-max=200000` or `echo "fs.file-max=200000" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.

- Check the user, group, and process settings for the maximum number of open file descriptors by executing the command `ulimit -Hn and ulimit -Sn`, where -Hn is the hard limit and -Sn is the soft limit. If the value is less than 200000, log on as a non-root user and increase the value by executing `ulimit -n 200000`. To permanently save this setting for the user, execute:

  ```
  echo "<user name> soft nofile 200000" >> /etc/security/limits.conf
  ```

  ```
  echo "<user name> hard nofile 200000" >> /etc/security/limits.conf
  ```

- Enter the host name of the machine on which you are installing products in the DNS of the network or in the file /etc/hosts.

- API Portal uses Elasticsearch 7.8.1. If you are going to install on a Linux system, the maximum map count checks that the kernel allows a process to have at least 262,144 memory-mapped areas. Go to the /etc/sysctl.conf file and set vm.max_map_count value=262144. Then restart the machine to make this change to take effect.

- If you are going to install on a Linux system, Install the fontconfig library.

Prepare for ApplinX Installation

- If you are going to install on a Windows system, and you want to develop .NET ApplinX Web applications in Visual Basic or C#, install Microsoft Visual Studio 2017 or 2019 and the Microsoft Visual J# .NET 2.0 Redistributable Package - Second Edition (x64). Microsoft Visual Studio does not have to be running when you install ApplinX.

  If you do not install Microsoft Visual Studio, you will not be able to install the ApplinX Visual Studio Add-in component, and the ApplinX .NET Framework component will have limited functionality; you will not be able to develop composite Web applications.

- If you are going to install on a Linux x86_64 system, install the rpm package compat-libstdc++ from your operating system distribution using the vendor instructions.

Prepare for CentraSite Installation (UNIX System)

- Make sure you have set sufficient user limits for the shell you use to start the installation and the product daemons. For example, the daemons for the CentraSite database and infrastructure will run out of memory if they are started from the shell with a low user limit for data. If your system policy allows it, Software AG recommends setting the value for coredump, data, file, memory, and threads to unlimited, and the value of nofiles to 8192. For more information about setting and displaying the ulimits, read the man page or ask your system administrator.
- Set the shmmax parameter to the memory (RAM) that is physically available on your machine. Otherwise you might have problems during CentraSite startup.

Prepare for EntireX Installation

- If you are going to install on a Linux x86_64 system, install the rpm package compat-libstdc++ from your operating system distribution using the vendor instructions.
- If you are going to install on a SUSE Linux Enterprise Server 11 SP1 system, and you have configured the system to use the Blowfish algorithm to encrypt user passwords, download and install patch 18 Aug 2011 - glibc 4944 from Novell Support. This patch fixes an operating system bug relating to user authentication. For detailed information, see Novell's SUSE Security Announcement: SUSE-SA:2011:035.
- If you are going to install EntireX on a UNIX system, make sure you have set sufficient user limits for the shell you use to start the installation and the product daemons. If your system policy allows it, Software AG recommends setting the value for coredump, data, file, memory, and threads to unlimited, and the value of nofiles to 8192. For more information about setting and displaying the ulimits, read the man page or ask your system administrator.
- If you are going to install EntireX on a UNIX NFS file system and want to do user authentication for EntireX Broker against the local operating system, you must mount the NFS file system with the suid option, which allows set-user-identifier or set-group-identifier bits to take effect.

Prepare for MashZone NextGen Installation (Linux System)

- Check the setting for shared memory (kernel parameter shmmax) by executing the command `sysctl -a | fgrep kernel.shmmax`. If the value is less than 629145600, log on as root user and increase the value by executing `sysctl -w kernel.shmmax=629145600` or `echo "kernel.shmmax=629145600" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.
- Check the settings for the system-wide maximum number of file descriptors (kernel parameter fs.file-max) by executing the command `sysctl -a | fgrep fs.file-max`. If the value is less than 200000, log on as the root user and increase the value by executing `sysctl -w fs.file-max=200000` or `echo "fs.file-max=200000" >> /etc/sysctl.conf`, then activate the new value by executing `sysctl -p`.
- Check the user, group, and process settings for the maximum number of open file descriptors by executing the command `ulimit -Hn and ulimit -Sn`, where -Hn is the hard limit and -Sn is the soft limit. If the value is less than 200000, log on as a non-root user and increase the value by executing `ulimit -n 200000`. To permanently save this setting for the user, execute:

  ```
  echo "<user name> soft nofile 200000" >> /etc/security/limits.conf
  ```

  ```
  echo "<user name> hard nofile 200000" >> /etc/security/limits.conf
  ```

- Enter the host name of the machine on which you are installing products in the DNS of the network or in the file /etc/hosts.

Prepare for Mobile Designer Installation

- Install an integrated development environment (IDE). Software AG recommends using the Eclipse provided with Software AG Designer.
- If you are going to use Mobile Designer separately from Software AG Designer, install Apache ANT 1.8 or higher using the instructions in the vendor documentation.

Prepare for My webMethods Server Installation (UNIX System)

Set sufficient user limits for the shell you will use to start the installation and the product daemons. For example, if you have multiple language packs installed for My webMethods Server, or if there are multiple products on the machine, the daemons for My webMethods Server will run out of memory if they are started from the shell with a low user limit for data. If your system policy allows it, Software AG recommends setting the value for coredump, data, file, memory, and threads to at least 32768, and the value of nofiles to 8192. For more information about setting and displaying the ulimits, read the man page or ask your system administrator.

Prepare for webMethods OneData Installation (Windows System)

Install Microsoft Visual C++ Runtime 2005 or higher in any directory on the machine that will host Locate.

Prepare for Optimize Analytic Engine Installation (UNIX System)

Set sufficient user limits for the shell you will use to start the installation and the product daemons. For example, the daemons for the Optimize infrastructure will run out of memory if they are started from the shell with a low user limit for data. If your system policy allows it, Software AG recommends setting the value for coredump, data, file, memory, and threads to at least 32768, and the value of nofiles to 8192. For more information about setting and displaying the ulimits, read the man page or ask your system administrator.

Prepare for Tamino Installation (Linux or UNIX System)

- Make sure you have set sufficient user limits for the shell you use to start the installation and the product daemons. For example, the daemons for the Tamino database will run out of memory if they are started from the shell with a low user limit for data. If your system policy allows it, Software AG recommends setting the value for coredump, data, file, memory, and threads to unlimited, and the value of nofiles to 8192. For more information about setting and displaying the ulimits, read the man page or ask your system administrator.

- Set the shmmax parameter to the memory (RAM) that is physically available on your machine. Otherwise you might have problems during Tamino startup.

Prepare for Zementis Predictive Analytics Installation

Set an environment variable with the name ADAPA_HOME to the absolute path to the directory where Zementis Predictive Analytics will store all uploaded artifacts. These artifacts include PMML files, custom resources, and the product license key file (zementis.license).

Note:
If you cannot or do not want to set the environment variable, you can go to the *new_Software AG_directory*/profiles/CTP/configuration directory after installation is complete, open the Zementis Predictive Analytics custom_wrapper.conf file, and add the line `set.ADAPA_HOME=%OSGI_CONFIGURATION_AREA%/adapa-folder`.

Prepare Database Connection Information

Some products require you to supply database connection information during installation. Keep in the mind the following:

- Most products use the DataDirect Connect JDBC driver. For information about options supported by this driver, see DataDirect Connect documentation, available on the Software AG Documentation website.

- Use the DataDirect Connect connection option MaxPooledStatements=35 on all database URLs except those for Trading Networks. This connection option improves performance by caching prepared statements. (Trading Networks caches its prepared statements using its own pooling mechanism).

- For DB2 database URLs:

- You must specify schema names using all uppercase letters. The tool you use to create database components later in this guide creates schema names in all uppercase letters, and DB2 is case sensitive for schema names.
- if your product will connect to a schema other than the default schema for the specified database user, you must specify these connection options in the database URL, where `AlternateID` is the name of the default schema that is used to qualify unqualified database objects in dynamically prepared SQL statements:

  ```
  ;AlternateId=schema;"InitializationString=(SET CURRENT
  PATH=current_path,schema)";MaxPooledStatements=35
  ```

- For Integration Server and Microservices Runtime, for MySQL Community Edition 5.7, you must specify connection options for the relaxAutoCommit, useLegacyDatetimeCode , and serverTimezone parameters. For example, you can provide these connection options as follows:

  ```
  jdbc:mysql://host:{3306|port}/databaseName?
  ```

  ```
  relaxAutoCommit=true&useLegacyDatetimeCode=false&serverTimezone=PST
  ```

- If the database user and password do not yet exist, specify the database user and password you will create after installation. You can use one database user for multiple database components, or you can use a different database user for each database component.

4 Preparing for Database Component Creation

Database Component Descriptions and Installation Requirements

A database component is a grouping of database objects that can be used by one or more products. For example, Integration Servers write service status data to the ISCoreAudit database component, and Monitor reads the data and displays it in its user interface.

This section lists the products that have database components and describes the database components to create. Each database component is named for the type of data that is written to it.

Note:
For detailed information on product concepts mentioned in the sections below, see the relevant product documentation.

ActiveTransfer Server Database Components

If you are going to install ActiveTransfer Server, create the ActiveTransfer database component. ActiveTransfer Server writes listener (port), server, user configuration, and other metadata to this database component. ActiveTransfer Server also writes run-time data, such as file transfer transaction information and audit data. If you are clustering ActiveTransfer Servers, create a single ActiveTransfer database component for the cluster to share. If you are not clustering, create an ActiveTransfer database component for each ActiveTransfer Server.

If you want to archive ActiveTransfer transaction data, create a separate schema or database, as appropriate for your RDBMS, in which to store the archived data. The ActiveTransferArchive database component installs stored procedures for archiving the data. The stored procedures need access to the production database tables listed in the section on archiving the database in *Managing File Transfers with webMethods ActiveTransfer*. If your organization will allow access to production database tables from the schema or database where the archived data resides, install the ActiveTransferArchive database component in the archive schema or database. If your organization will not allow such access, install the ActiveTransferArchive database component in the production schema or database, and give the location of the archive schema or database during the archive process.

Note:
If you are using Oracle, you might be able to use partitioning for the ActiveTransferArchive database component. Contact Software AG Professional Services for information.

API Gateway Database Component

API Gateway comes with a data store in which it stores core data, events, and metrics. If you want to store information about runtime event invocations for analytics and auditing purposes, for either one API Gateway or a cluster of API Gateways, create one APIGatewayEvents database component.

CloudStreams Database Component

If you are going to install CloudStreams, create the CloudStreamsEvents database component. CloudStreams Server writes lifecycle (startup/shutdown), error, policy violation, monitoring, performance metric, and, optionally, transaction events to the CloudStreamsEvents database component. CloudStreams Analytics reads the events data and displays it using the MashZone NextGen dashboard. If you have multiple CloudStreams Server instances, create a single CloudStreamsEvents database component for them to share, regardless of whether they are clustered or not clustered.

Integration Server or Microservices Runtime Database Components

The table below describes the types of data Integration Server or Microservices Runtime can persist.

Database Component	Types of Data	When Data is Written
ISInternal	Service results, scheduled tasks, client certificate mappings, run-time data for pub.storage services, guaranteed delivery transactions, trigger joins, active OpenID requests, WS-ReliableMessaging runtime data, and configuration and runtime data for OAuth and Account Locking.	You are using the features listed in the **Types of Data** column
ISCoreAudit	■ Error, guaranteed delivery, service, security, and session audit data.	The audit logger for the type of data is enabled
	■ Documents that are in doubt, have failed, or have exhausted trigger retries.	You are using triggers
CrossReference	Cross-referencing data for publish-and-subscribe solutions.	You are using publish-and-subscribe solutions
DocumentHistory	Document history data for exactly-once processing in publish-and-subscribe solutions. Integration Server uses the data to detect and reject duplicate documents.	You are using exactly-once processing
DistributedLocking	Information that coordinates access to resources across distributed servers and processes.	Executing services in the pub.storage folder.

External RDBMS versus Embedded Database

You must use an external RDBMS with Integration Server or Microservices Runtime if you are going to:

- Cluster server instances.
- Install in a production environment with high transaction volumes or large datasets. The definition of "high" and "large" depends on the amount of memory available to your JVM, the speed of your CPU, and the kind of work your application is doing (for example, more processing intensive, or more IO intensive).
- Need a database that is scalable, visible, and highly reliable, and that has robust management and recovery tools.
- Write document history data.
- Have services log their input pipelines.
- Use triggers for joins.
- Use pub.storage for high-volume storage.
- Write Process Audit Log and Process Engine data for business processes.
- Use Business Rules with Auditing or Hot Deployment.
- Use Monitor.

If you are not going to do any of the above, and your database demands are low, your needs might be met by using the embedded database. For example, you can use the embedded database when you are using Integration Server or Microservices Runtime as a runtime for adapters, eStandards Modules, API Gateway, or Deployer.

Using the Embedded Database

Integration Server and Microservices Runtime use Derby, a file-based database, as their embedded database. When you use the embedded database, Integration Server or Microservices Runtime writes IS Internal and Cross Reference data to that database, and writes IS Core Audit Log data to files.

Important:
If one database file gets corrupted, the entire database might become unusable. Software AG therefore recommends backing up the \IntegrationServer\instances\instance_name\db directory for Integration Server or the *Software AG_directory* \db directory for Microservices Runtime periodically so you will be able to return to the last known good state if necessary.

If you later want to write these types of data to an external RDBMS instead, you will need to create the necessary database components in the RDBMS (see “ Database Component Descriptions and Installation Requirements” on page 28) and configure Integration Server or Microservices Runtime to write to them (see “Connecting Products to Database Components” on page 70). You must also complete the steps for switching from the embedded database to an external RDBMS (see *webMethods Integration Server Administrator’s Guide*).

Using the External RDBMS

When you use an external RDBMS, create the ISCoreAudit, ISInternal, and DistributedLocking database components in the RDBMS. You must create the ISInternal and DistributedLocking database components in the same schema or database, as appropriate for your RDBMS. Also create other database components as needed. For example, if you are using publish-and-subscribe solutions and exactly-once processing, you will need the CrossReference and DocumentHistory database components; if you are running business processes, you will need the ProcessAudit and ProcessEngine database components. For complete information about the database components you will need to create, see “ Database Component Descriptions and Installation Requirements” on page 28.

If you are clustering server instances, create an ISCoreAudit database component, an ISInternal database component, and a DistributedLocking database component in the same schema or database, as appropriate for your RDBMS, for the cluster to share. Also create a CrossReference database component and a DocumentHistory database component for the cluster to share.

If you are not clustering server instances, create an ISInternal database component and a Distributed Locking database component for each server instance. Also create the ISCoreAudit, CrossReference, and DocumentHistory database components; you can create one of each of those database components for all server instances to share, or you can create each of those database components for each server instance.

Archiving the ISCoreAudit Database Component

If you want to archive data from the ISCoreAudit database component (called the "source" database component in this section), create the Archive database component. The Archive database component includes a copy of the source database components. You must create the Archive database component in the same type of RDBMS and on the same database server as the source database components, but in a different schema or database, as appropriate for your RDBMS, than the source database components.

The OperationManagement database component is automatically created when you create the Archive database component. It provides a common method for configuration, logging, and error handling for the Archive database component.

If you are also going to archive ProcessAudit data, you can use the same Archive database component for both the ISCoreAudit and ProcessAudit database components.

Mobile Support Database Component

If you are going to install the Mobile Support package on Integration Server, create the MobileSupport database component. Mobile Support writes data used for synchronizing mobile solutions, and metadata about that data, to this database component.

My webMethods Server Database Components

Create the MywebMethodsServer database component if you are going to install any of the following:

- My webMethods Server, with or without Task Engine, the user interface for Monitor, Optimize, or Trading Networks, or the Central User Management package for Integration Server
- Task Engine on Integration Server
- The Central User Management package and the Common Directory Service Support runtime library for Integration Server, to use with the Integration Server Administrator user interface (preview feature).

My webMethods Server writes data about deployment, configuration, security, portal pages, and run-time operations to this database component. It uses this data to manage the My webMethods user interfaces (for example, the user interfaces for Monitor, Optimize, and Trading Networks) and to support central user management in Integration Server.

The Task Engine on My webMethods Server and the Task Engine on Integration Server write task status, task distribution, and business data. The Task Engine on My webMethods Server displays this data in its user interface in My webMethods. The Task Engine on Integration Server displays this data in its gadgets in Business Console and in a Task Engine browser-based user interface.

If you are clustering My webMethods Servers, create a single MywebMethodsServer database component for the cluster to share. If you are not clustering, create a MywebMethodsServer database component for each My webMethods Server.

The CentralConfiguration database component is automatically created when you create the MywebMethodsServer database component. If you are using Optimize, the CentralConfiguration database component stores the Optimize configuration information you enter in the Central Configurator interface in My webMethods. The CentralConfiguration database component is not automatically *dropped* when you drop the MywebMethodsServer database component. If you want to drop the CentralConfiguration database component, you must do so manually.

webMethods OneData Database Components

If you are going to install webMethods OneData, create the webMethods OneData database components.

- webMethodsOneDataMetadata: webMethods OneData writes internal configuration data to this database component.
- webMethodsOneDataWorkArea: webMethods OneData users create data objects and work-in-progress data values to this database component.
- webMethodsOneDataReleaseArea: Contains the same data objects as the webMethodsOneDataWorkArea database component, and approved data values. Data values are deployed from the webMethodsOneDataReleaseArea database component.

You must create each webMethods OneData database component in a separate schema or database, as appropriate for your RDBMS, and use a different database user for each database component. You must create the webMethodsOneDataMetadata database component first, then the webMethodsOneDataWorkArea database component, then the webMethodsOneDataReleaseArea database component.

Optimize Database Components

If you are going to install Optimize, create the Analysis and ProcessTracker database components.

Optimize Analytic Engines write computed analytics and process and monitoring data received from Infrastructure Data Collectors and Web Service Data Collectors to the Analysis database component. The Optimize user interface displays the data. Create a single Analysis database component for all Optimize Analytic Engines to share. If you are going to use root cause analysis for business processes, install the Analysis and ProcessTracker database components in the same schema or database, as appropriate for your RDBMS.

Optimize Analytic Engines write business and process status data received from processes that are not orchestrated by the Process Engine to the ProcessTracker database component. The Optimize user interface displays the data. Monitor reads process status data from this database and displays it in the Monitor interface in My webMethods. Create a single ProcessTracker database component for all Optimize Analytic Engines to share. If you are going to use root cause analysis for business processes, install the ProcessTracker and Analysis database components in the same schema or database, as appropriate for your RDBMS.

The DataPurge and DatabaseManagement database components are automatically created when you create the Analysis or ProcessTracker database component. The DataPurge database component provides a common method for purging data from the Analysis and ProcessTracker database components. The DatabaseManagement database component provides core routines for data purge functionality.

The OperationManagement database component is automatically created when you create the Analysis or ProcessTracker database component. It provides a common method for configuration, logging, and error handling for those database components.

Process Engine Database Components

If you are going to install Process Engine, create the ProcessEngine and ProcessAudit database components.

Process Engines write process execution data for processes they orchestrate to the ProcessEngine database component. If you are distributing business process steps, you cluster the Process Engines that orchestrate the steps. Create a single ProcessEngine database component for the cluster to share. If you are not clustering, create a ProcessEngine database component for each Process Engine.

Process Engine and the following write to the ProcessAudit database component:

- Process Engines write process audit data for business processes they orchestrate.
- Task Engines write task audit data for task steps orchestrated by Process Engine.
- Third-party products can write process execution data.

The following read from the ProcessAudit database component:

- Optimize Analytic Engines read process data so they can analyze capabilities such as transition duration and write data about analysis-enabled processes, then displays this data in the Optimize interface in My webMethods.
- Monitor reads process data and displays it in the Monitor interface in My webMethods, where you can track the status of process instances, view logged values, and, in some cases, resubmit process instances.
- Business Console reads and displays process data so you can monitor process instances in real time.

If you are distributing business process steps, you cluster the Process Engines that orchestrate the steps. Create a single ProcessAudit database component for the cluster to share. If you are not distributing business process steps, and therefore not clustering Process Engines, you can create either a ProcessAudit database component for each Process Engine or a single ProcessAudit database component for all Process Engines to share.

If you are using Process Engines, Task Engines, Optimize Analytic Engines, or some combination of these, create a single ProcessAudit database component for all to share.

Note:
You might be able to use partitioning for the ProcessAudit database component. Contact Software AG Professional Services for information.

Archiving the ProcessAudit Database Component

If you want to archive data from the ProcessAudit database component (called the "source" database component in this section), create the Archive database component. The Archive database component includes a copy of the source database components. You must create the Archive database component in the same type of RDBMS and on the same database server as the source database components, but in a different schema or database, as appropriate for your RDBMS, than the source database components.

The OperationManagement database component is automatically created when you create the Archive database component. It provides a common method for configuration, logging, and error handling for the Archive database component.

If you are also going to archive ISCoreAudit data, you can use the same Archive database component for both the ProcessAudit and ISCoreAudit database components.

Rules Engine Database Component

If you are going to install the Rules Engine, create the BusinessRules database component. As each Rules Engine instance starts, it registers itself in this database component and stores information about deployed projects and the state of business rules instances. When you modify a business rule, the hot deployment functionality in the Business Rules user interface enables you to deploy changes to all of the instances registered in this database component. If you install multiple Rules Engine instances, create a single BusinessRules database component for the instances to share.

Note:

You do not need to create the BusinessRules database component since the Rules Engine does not need a database connection if it is excluded from hot deployment and does not use auditing. For more information on hot deployment and writing business rules auditing information, see the webMethods Business Rules documentation.

Trading Networks Server Database Components

If you are going to install Trading Networks Server, create the TradingNetworks database component. Trading Networks Server writes metadata (partner profiles, trading partner agreements, document types, processing rules, and partner profile groups) and run-time data (documents, document content parts, attributes, and delivery information) to the TradingNetworks database component. Trading Networks Server also logs entries about document processing steps. The Trading Networks user interface displays the data. If you are clustering Trading Networks Servers, create a single TradingNetworks database component for the cluster to share. If you are not clustering, create a TradingNetworks database component for each Trading Networks Server.

If you want to archive Trading Networks data, create the TradingNetworksArchive database component. Create the TradingNetworksArchive database component in the same schema or database, as appropriate for your RDBMS, as the source TradingNetworks database component.

Note:
If you are using Oracle, you might be able to use partitioning for the TradingNetworksArchive database component. Contact Software AG Professional Services for information.

Database Drivers for DB2, MySQL Enterprise Edition, Oracle, PostgreSQL, and SQL Server

Software AG products that support DB2, MySQL Enterprise Edition, Oracle, PostgreSQL, or SQL Server use the DataDirect Connect JDBC driver to communicate with those RDBMSs. Oracle and SQL Server use DataDirect Connect JDBC driver 6.x and the other RDBMSs use DataDirect Connect JDBC driver 5.x.

The products come with the client component of this driver; it is a JDBC native-protocol driver that does not have a server component. The products execute DDL commands at install time and sometimes design time for each RDBMS with which they interact. For information about the DataDirect Connect JDBC driver, see the DataDirect Connect documentation, available on the Software AG Documentation website.

Important:
You must use the provided DataDirect Connect JDBC driver with your products and the RDBMSs listed above unless the documentation for a product states that it supports another driver.

Prepare for Database Component Creation in MySQL

Software AG products that support MySQL Community Edition use the MySQL Community Edition database driver to communicate with the RDBMS. For information about the driver, see the vendor documentation.

Note:
If you experience any issues with the MySQL native driver, or with fixes, service packs, or patch-bundles for the driver, report them directly to the vendor. Software AG is not responsible for addressing driver-related issues.

Install the MySQL Community Edition Database Driver for Integration Server, Microservices Runtime, and Hosted Products

Download the MySQL Community Edition driver to the *Software AG_directory* \IntegrationServer\lib\jars\custom directory.

Install the MySQL Community Edition Database Driver and Edit the MySQL Configuration File for All Other Products

1. Download the MySQL Community Edition driver to the *Software AG_directory* \common\lib\ext directory.

2. Rename the MySQL Community Edition driver jar to mysql-connector-java.jar.

3. Go to the `Software AG_directory`\common\db\bin directory, open the setEnv.bat file, and add the MySQL Community Edition driver jar to the classpath, as follows:

   ```
   set CLASSPATH=%CLASSPATH%;%DCI_HOME%\..\lib\ext\mysql-connector-java.jar
   ```

4. Open the MySQL configuration file.

   a. Remove the values `NO_ZERO_IN_DATE` and `NO_ZERO_DATE` from the sql_mode variable.

   b. If MySQL is on a Linux system, set the parameter `lower_case_table_names=1`. If MySQL is on an AWS RDS system, set the parameter `log_bin_trust_function_creators=1`.

   c. Add the parameter `transaction-isolation=READ-COMMITTED`.

Prepare for Database Component Creation for All RDBMSs

Database Administrator Account

You need DBA access only if you want to use the Database Component Configurator to create the necessary database users and storage objects.

Your products are designed to use the database users you will create for the database components using the instructions in this guide.

Character Set and Sort Order

Your products are globalized and support Unicode. Software AG strongly recommends choosing a Unicode encoding for your database and the most appropriate sort order for your environment. A database character set determines which languages a database can represent. Database sort order determines collation and comparison behavior.

The sections below list the most appropriate Unicode character encoding and sort order for each RDBMS that your products support. If you want to use a different character set or sort order than recommended below, consult your database administrator and your RDBMS vendor's documentation so you can carefully choose a database character set that supports the languages your data is in.

If you use the Database Component Configurator to create your database components, you can check whether the selected RDBMS is configured for the Unicode character set. If the RDBMS does not support Unicode, the configurator lists the character set the RDBMS does support.

Important:
You must set character set and sort order before creating storage.

DB2

Database schemas for DB2 use character data types. DB2 supports UTF-8 for character data types and UTF-16 for graphic data types.

The table below lists the character sets and sort order recommended by Software AG.

For	Software AG Recommendation
Character set	CCSID 1208 (UTF-8) My webMethods Server requires this character set.
Graphic Character Set	UTF-16
Sort order	IDENTITY_16BIT This sort order ensures the same sorting result for both character and graphic data types.

You can check the database configuration using the GET DATABASE CONFIGURATION command.

MySQL Community Edition and Enterprise Edition

The server character set and collation are used as default values if the database character set and collation are not specified in CREATE DATABASE statements. They have no other purpose.

You can determine the current server character set and collation settings from the values of the character set server and collation server system variables. You can change these variables at runtime.

The table below lists the character set and sort order recommended by Software AG.

For	Software AG Recommendation
Character set	UTF-8
Collation	utf8_general_ci

You can check the database configuration using the SHOW VARIABLES LIKE command.

Oracle

Database schemas for Oracle use character data types. For character data types, Oracle supports the UTF8 and AL32UTF8 Unicode encodings. While UTF8 is CESU-8 compliant and supports the Unicode 3.0 UTF-8 Universal character set, AL32UTF8 conforms to the Unicode 3.1 or higher UTF-8 Universal character set. For nchar data types, Oracle supports the AL32UTF8 and AL16UTF16 Unicode encodings. The supported Unicode version for AL32UTF8 depends on the Oracle database version. Oracle database schemas for your products do not have linguistic indexes.

The table below lists the character sets and sort order recommended by Software AG.

For	Software AG Recommendation
Character set	AL32UTF8
Nchar character set	AL16UTF16
Sort order	Binary

You can check database configuration and session settings by viewing the SYS.NLS_DATABASE_PARAMETERS or V$NLS_PARAMETERS parameter.

PostgreSQL

PostgreSQL uses UTF-8 encoding by default.

SQL Server

Database schemas for SQL Server use nchar data types. SQL Server provides support for UTF-16 through its nchar data types. Since nchar data types are always in UTF-16, you do not have to perform any special database configuration.

Some products, such as Process Engine, require a double-byte character set (DBCS). Choose the most appropriate code page for your environment as a database character set.

The table below lists the character sets and sort order recommended by Software AG.

For	Software AG Recommendation
Character set	The appropriate encoding for the languages your data is in.

For	Software AG Recommendation
Nchar character set	UTF-16
Sort order	Any case-insensitive collation type. If you do not choose a case-insensitive sort order, you will not be able to create some database components in SQL Server.

You can check the database configuration using the sp_helpdb *database* stored procedure.

Page and Block Size

The table below provides the page and block sizes specified to use for each type of RDBMS.

RDBMS	Required Page and Block Size	Default
DB2	32K page size	4K
MySQL	16K page size	16K
Oracle	8K page/block size	8K
PostgreSQL	8K page/block size	8K
SQL Server	8K page/block size	8K

Set Database Options

You must set database options for the RDBMSs below.

MySQL

Set the parameter default_time_zone='+00:00' in the MySQL configuration file for the [client] and [mysqld] sections.

Oracle

For your products to function properly, you must set the NLS_LENGTH_SEMANTICS initialization parameter to BYTE.

Important:
Your products use BYTE rather than CHAR semantics because BYTE is the default configuration of Oracle database and is used by most Oracle users. If you try to use your products with CHAR semantics, unexpected results might occur. For example, since CHAR semantics allow storage of longer data than BYTE semantics (for Japanese, varchar(12 byte) takes four characters in UTF8 while varchar(12 char) takes 12), using CHAR semantics could cause the buffer in some products to overflow.

SQL Server

Software AG recommends the following database options for your products to function properly:

```
ALTER DATABASE database_name SET ALLOW_SNAPSHOT_ISOLATION ON;
ALTER DATABASE database_name SET READ_COMMITTED_SNAPSHOT ON;
```

In addition, enable Named Pipes and TCP/IP protocols using the SQL Server Configuration Manager.

By default, SQL Server uses case-insensitive collations. If you create the My webMethods Server, Integration Server, or Trading Networks database component, do not change this option from the default; My webMethods Server,Trading Networks, and Integration Server do not support case-sensitive collations.

webMethods modules are not tested for case sensitivity. Setting different collation in the database might or might not work, but it is not tested and not supported.

5 Installing Products, Creating Database Components, and Connecting Products to Database Components

Installing Products and Creating Database Components Using Command Central

Follow the instructions in *Software AG Command Central Help* to install your new products and fixes and create your database components using Command Central.

Installing Products Using Software AG Installer and Creating Database Components Using Database Component Configurator

Installing Products Using Software AG Installer

Recommendations

- Software AG strongly recommends that you create installation images of your products and store them on your internal network. Create an image for each operating system on which you plan to install that is covered by your license agreement. Storing images enables you to ensure consistency across installations over time; the installer provides only the latest product releases, while installation images contain the product releases you download into them. In addition, installing from images helps you reduce WAN traffic and improve installation speeds. For instructions on creating an installation image, see *Using Software AG Installer*.
- Unless otherwise stated for a specific product, do not install products from this release into a Software AG directory that contains products from any other release. Unless otherwise stated for a specific product, you cannot mix products from different releases in the same installation directory; if you do so, you will experience problems or be unable to access functionality.
- When you need multiples of the same release of a product on one machine, Software AG recommends that you create multiple instances of the product in the same installation rather than creating multiple installations of the same product on the machine. Most products now offer this functionality. The valid use cases for creating multiple installations on the same machine are when those installations are from different releases (for example, side-by-side upgrade), different environments (for example, production environment on fix level 1 and staging environment on fix level 2), or different projects (for example, installations used by different departments within an organization).

  > **Note:**
  > If you are using a virtualization layer such as VMWare, the above use cases are typically handled using different VM instances on a single powerful VM server, with a single installation on each VM instance.

Before Installing Using Software AG Installer

Gather Information

As you go through the Software AG Installer, you will be asked for various types of information, such as product license files and database connection parameters. Gather this information before starting the installer.

Save and Verify Product License Files

Software AG sends product license files with your installation message; save the license files in a directory whose name and path do not contain any spaces.

When you provide the license files during installation, the installer performs the validations below. You will only be able to install a product if its license file passes these checks. The products will repeat these checks at startup.

- The license file must be the correct one for the product.
- The license file must not have been changed.
- The license file must not have expired.
- The product must be licensed on the operating system on which you are installing.

Read and Perform Required Actions in *Using Software AG Installer*

Read and perform the required actions in *Using Software AG Installer*. For example, for Windows create a user account for installation that has the proper privileges; for UNIX, create a non-root user account.

API Gateway and Developer Portal Requirements

API Gateway and Developer Portal come with a data store, called API Data Store, in which they store core data, events, and metrics. The data store uses Elasticsearch 8.2.3, which does not support as many operating systems as API Gateway or Developer Portal. If you install API Gateway or Developer Portal on an operating system that is not supported by the data store, install the data store on a supported operating system and then configure API Gateway or Developer Portal to use the data store. For configuration instructions, see the product documentation.

If you do install the data store on a different operating system than Developer Portal, go to the *Software AG_directory*/profiles/CTP/configuration directory, open the dpo_wrapper.conf file, and specify the URL for the data store on the -Dspring.elasticsearch.rest.uris property.

CentraSite Requirements

- You can install the CentraSite Registry Repository and Application Server Tier on different machines, in different directories on the same machine, or in the same directory but at different times (that is, in two different runs of the installer). Whichever approach you follow, you must always install the Registry Repository first.

If you are going to install multiple instances of CentraSite components on the same machine:

- Each instance of the CentraSite Registry Repository, Application Server Tier, and Platform Manager must use unique ports.
- Software AG recommends that you have all Application Server Tier instances on the machine running before you install a new Application Server Tier instance, so the installer can warn you if a port is already in use.
- All Registry Repository instances on the machine must be shut down before you install a new Registry Repository instance. However, because the instances are not running, the installer will not be able to warn you if the default port is in use; you will have to find and assign a unique port yourself.

Deployer and Asset Build Environment Requirements

If no firewall separates your integration, staging, and production environments, install one Deployer to handle all environments. If a firewall separates your integration and staging environments from your production environments, install one Deployer to handle your integration and staging environments and a separate Deployer to handle your production environments. The Deployer host machine must have access to all source servers from which you want to deploy, and all target servers to which you want to deploy. Restrict access to the Deployer host machine to only those people who need to use Deployer, such as release managers, infrastructure engineers, system administrators, and operators.

Install Asset Build Environment on development machines on which assets are created so you can verify successful builds before checking code into your VCS. For centralized builds, install Asset Build Environment on a separate machine, known as a continuous integration server, where you will use it to build assets into *composites* that can be deployed. Restrict access to the Asset Build Environment host machine to only those people who need to use Asset Build Environment, such as release managers, infrastructure engineers, system administrators, and operators.

Install Products Using Software AG Installer

Start the Installer and Provide General Information

The table below explains how to start the installer.

If you are installing...	Do the following...
MashZone NextGen on a Windows system	Install using the Windows Administrator user account. To do so, instead of double-clicking the installer .exe file to start the installer, right-click the .exe file and click **Run as administrator**. Other user accounts do not work, even if they were assigned administrator privileges.
All other products	Follow the instructions in *Using Software AG Installer* to start the installer and provide general information such as proxy server, release to install, installation directory, and how to use the product selection tree.

If you are installing CentraSite, on the installation directory panel, the installer auto-detects a host name or IP address for the machine on which you are running the installer. If you are installing CentraSite, you might need to change that value, as follows:

- If you install the Application Server Tier and Registry Repository on different machines, the value when you install the Registry Repository must specify a host name or IP address that the Application Server Tier can use to access the Registry Repository.
- If you are installing CentraSite in the cloud, the installer auto-detects the internal IP address of the cloud instance. If you want CentraSite to be accessible from outside the cloud instance (for example, from another cloud instance running other products), replace the auto-detected value with an externally accessible host name or IP address.

Choose Products to Install

On the product selection tree, select the products to install. The installer lists all products you have licensed that are supported on the machine on which you are installing.

Important:
The installer does not distinguish among versions (Windows) or flavors (UNIX) of an operating system. Make sure you install products only on the supported versions or flavors listed in the *System Requirements for Software AG Products*.

Note:
Software AG Installer offers a feature that helps you create typical development installations that enable you to perform a certain task (for example, API management). When you select a typical development installation, the installer automatically selects all products that make up that installation. You can select more products or deselect products if desired.

The table below provides important information you will need to know to correctly install certain products.

Product	Important information needed for correct installation
Products that can run on either Integration Server or Microservices Runtime	The installer will ask which host server you want to use to host the product. An example of such a product is Deployer.
Apama	The Apama typical development installation does not include Apama Platform Manager Plug-in. If you want to manage your Apama installation from Command Central, manually select the plug-in when selecting the Apama components to install. **Note:** Command Central integration with Apama is deprecated and will be removed in a later release.

Product	Important information needed for correct installation
API Gateway	The license for API Gateway allows only partial functionality of its host Integration Server. Install API Gateway on its own host server. Do not install other products on that host server or those other products might not work properly.
CloudStreams	The license for CloudStreams allows only partial functionality of its host server (that is, Integration Server or Microservices Runtime). Install CloudStreams on its own host server. Do not install other products on that host server because those other products might not work properly.
Integration Server	If you select packages to install with Integration Server, the installer will copy the packages into a package repository in the product installation. The installer will also create a server instance and will ask whether to install all packages in the package repository on the server instance. After installation, you can create additional server instances and install packages from the package repository on those server instances as well as on the server instance created during installation. For instructions, see the *webMethods Integration Server Administrator's Guide*.
	If you install the Common Directory Services component for use with the Integration Server Administrator user interface (preview feature), install the Central User Management package and the Common Directory Service Support Integration Server runtime library on Integration Server.
My webMethods Server	If you install My webMethods Server, and later install Business Console, Task Engine, or My webMethods user interfaces in a separate run of the installer, the installer will install those components on all My webMethods Server instances in the target installation directory.
Optimize	You must also install a package called Optimize Support on Integration Server or Microservices Runtime. This package provides data about business processes and KPIs to Optimize. If you install the package on Microservices Runtime, data is provided for unorchestrated business processes only; data is not provided for orchestrated business processes or events.

Choose Default Password for Products

On the administrator password panel, enter a default product administrator password for the products you are installing, and choose whether to require the password to be changed on first product login.

Exceptions:

- webMethods OneData does not support this panel and will ignore inputs on it, but you must still enter a password in order to continue with installation.
- Zementis Integrated Server does not support the choice on this panel to require password change on first login and will ignore this choice if selected.

Important:

Choose Language Packs to Install

For information on the language packs, see the international operating environments appendix in this guide.

The table below provides important information you will need to know to correctly install certain products.

Product	Important information needed for correct installation
CentraSite	If you select the CentraSite Application Server Tier language pack, you must also select the CentraSite Shared Files language pack.
Integration Server	If you select language packs to install with Integration Server, the installer will copy those language packs into the package repository and will always install them on the server instance. After installation, you can create additional server instances. You can install language packs from the package repository on those server instances. For instructions, see the *webMethods Integration Server Administrator's Guide*.

Install Latest Updates

If you are installing in a new, empty directory, and there are updates for the products you are installing, the installer displays the update selection tree after the language pack selection tree. Install the latest updates on you products. For information about each update, see the accompanying readme file.

Note:
If manual steps are required before a particular update can be installed, that update will be unavailable. You will have to run the Software AG Update Manager to install the update after installation is complete.

Supply Product Configuration Information

Most product panels or prompts are self-explanatory. This section provides information for those products that require additional explanation.

Important:
Make sure all ports you specify on panels or in response to prompts are not already being used by other products on your machine. The installer cannot detect whether ports are in use when products are shut down, and the shutting down of products is a requirement for running the installer. See the list of default ports used by Software AG products.

Note:
When you install on a UNIX system, the installer might ask whether to use sudo and related scripts. If you are creating or installing from an installation script, the option to use sudo is not

available because the installer does not store the sudo password in scripts, for security reasons. Instead you will perform certain tasks after installation as described later in this chapter.

Apama

If you are installing Apama Server, the installer asks for a license file. If you provide one, the installer will copy it into the Apama work directory. If you do not provide one, the Apama correlator will run with reduced capabilities. See the Apama documentation for information on running without a license or on providing a license after installation.

If you are installing the Apama Server or Dashboard Viewer, the installer asks for the path to the work directory in which Apama will store its user-created content, such as log files. You can specify the following as the work directory:

- A directory that does not exist (recommended). The installer will create the directory, then populate it with standard files and directories.
- A non-empty directory. The installer will not write standard files and directories to the directory, to avoid overwriting existing user content.

You can include non-English characters in the work directory location, but they must be within the set of characters from the operating system's default character encoding.

API Portal

The table below shows the system configurations you can select for API Portal to handle your expected usage needs.

System Configuration	Memory Consumption	Maximum Concurrent Users
Small	1 GB	50
Medium	4 GB	100
Large	8 GB	400

EntireX

The table below explains the fields that are not self-explanatory on the EntireX installer panel.

Field	Entry
License file	Full path to the EntireX license file. **Note:** If you are also installing NaturalONE, the NaturalONE license file appears in the field, and EntireX uses that license.

Field	Entry
Application Monitoring Data Collector port, and **Autostart**	■ Port to use for the Application Monitoring Data Collector. ■ Whether the installer should start the Data Collector automatically after installation and turn on the Autostart option so the the Data Collector will start automatically whenever you start your system.
EntireX Broker Administration Service port, and **Autostart**	When you install EntireX Broker: ■ Port for the Broker Administration Service. ■ Whether the installer should start the default Broker after installation and turn on the Autostart option so the Broker will start automatically whenever you start your system.

Integration Server or Microservices Runtime

If you are installing API Gateway, specify the license file for API Gateway, not the license file for the host Integration Server. Select **Embedded database**.

If you are installing CloudStreams, specify the license for CloudStreams, not the license for the host Integration Server or Microservices Runtime.

If you are installing Local Version Control Integration, specify the license for Local Version Control Integration.

If you are installing Deployer, select **Embedded database**.

To determine whether to use an embedded database or external RDBMS in all other cases, see "Preparing for Database Component Creation" on page 27.

webMethods OneData

The table below explains the fields on the webMethods OneData installer panel.

Field	Entry
Repository ID	webMethods OneData provides a repository in which to store your master data. Provide an ID for the repository (for example, DevRepo). The ID will be used internally in webMethods OneData.
Repository name	Provide a name for the repository (for example, OneData Development Repository). The name will appear in the webMethods OneData user interface.
Connection Prefix	webMethods OneData has three storage areas - Metadata, Work Area, and Release Area. The prefix you specify will be added to the schema connection names (that is, *prefix*_md, *prefix*_wa, and *prefix*_ra). If you override the default value for this field, you will have to edit the database connections for the Work Area and Release Area after you start webMethods OneData (see the webMethods OneData documentation).

Field	Entry
URL	The URL you specify must include the option catalogOptions=1 so that webMethods OneData can retrieve database metadata information for various operations (for example, jdbc:wm:oracle://localhost:1521;serviceName=XE;catalogOptions=1)
Database user and **Password**	Provide a different database user and password for each of the three schemas.
Schema name	You must specify schema names if you use a SQL Server RDBMS.

Universal Messaging

The table below explains the fields on the Universal Messaging installer panel when you choose to install a Universal Messaging realm server alone or with Enterprise Manager or Template Applications.

Field	Entry
License file	The contents of this field varies as explained below. ■ If you have the standard Universal Messaging license, specify the full path to that license file. ■ If you do not have a Universal Messaging license file, leave the field blank. You can use Universal Messaging on a trial basis for up to 90 days. If you obtain a Universal Messaging license file after installation, you can add it at that time. For instructions, see the Universal Messaging documentation.
NHP interface binding	If your machine has multiple IP addresses and you want Universal Messaging to listen to a specific one, select that IP address from the list.
NHP interface port	The installer sets up a bootstrap Universal Messaging interface to which all sample Universal Messaging applications and tools connect by default. Specify the port to which to bind the bootstrap interface.
Realm server name	Name for the realm server instance. If you are also installing an Enterprise Manager instance, Template Applications instance, or both, those instances will also use this name. **Note:** Different types of instances can have the same name. However, the name you supply here must be unique among instances of the same type (that is, among instances of realm servers).
Data directory	Full path to the directory in which to store the data for the realm server instance.

Field	Entry
	Note: If you use a non-default location for the data directory, you will have to manually migrate the data directory when you upgrade Universal Messaging. If you install multiple realm server instances on the same machine, use a different data directory for each instance.
Configuration	Initial configuration for the Universal Messaging realm server. Select the configuration for webMethods suite use cases.

The table below explains the fields on the Universal Messaging installer panel when you choose to install Universal Messaging Enterprise Manager, Template Applications, or both, without also choosing to install a Universal Messaging realm server.

Field	Entry
Realm name/client profile	Name for the Enterprise Manager instance, Template Applications instance, or both. You might want to use the name of the realm server instance to which the tools will connect. The list shows the names of any realm server instances that already exist in the installation directory, but you can also type a name. **Note:** Different types of instances can have the same name. However, the name you supply here must be unique among instances of the same type (that is, among instances of Enterprise Managers, and among instances of Template Applications).
Realm server URL (RNAME)	URL for the realm server instance to which the Enterprise Manager instance, Template Applications instance, or both should connect. You can specify a realm server that is already installed or that will be installed later.

After Installing Using Software AG Installer

Manage Products Using Command Central

Software AG Installer automatically installed Command Central's agent, Platform Manager, with your products. This enables you to manage the products remotely after installation from one location using Command Central. For instructions on managing products from Command Central, see *Software AG Command Central Help*.

Install Latest Updates if Not Available During Installation

If you installed your products in a new, empty directory, and there are updates for the products you installed, the installer displayed an update selection tree. However, if manual steps are required before a particular update can be installed, that update was unavailable. In addition, if you installed your products into a directory that already contained products, the installer did not offer the update selection tree.

In both cases you will need to install the latest updates using Command Central or the Software AG Update Manager. For instructions, see *Software AG Command Central Help* or *Using Software AG Update Manager*, respectively.

Start, Configure, and Customize Products

For instructions on starting, configuring, and customizing products, see the product documentation.

If you are using Software AG Metering, see the *Software AG Infrastructure Administrator's Guide* for configuration instructions.

For instructions on registering UNIX daemons, see the next section.

For instructions on configuring BigMemory Max when you are using it with webMethods products, see *Using BigMemory with webMethods Products*.

If you installed a Zementis Standalone Server, product documentation is in the *Software AG_directory*\Zementis\adapa-app\documents. For instructions on setting up the server, see the *Zementis Deployment Guide*.

Important:
If any product you installed has a default password, you should change that password as soon as possible. For instructions, see the product documentation.

Register Daemons and Set File Permissions (UNIX System)

If you choose to register daemons in the installer panels, but you created a script or installed from a script, or if you were asked whether to use sudo and you chose not to, the installer was not able to register the daemons and set file permissions. Perform these tasks now by running the script *Software AG_directory*/bin/ afterInstallAsRoot.sh with root permission (that is, call the script using the root user account or the sudo utility).

Note:
The afterinstallAsRoot.sh script only registers the daemons you selected in the installer panels. If you want to change or add daemon registration for any product at a later time, see "Registering Daemons to Automatically Start and Shut Down Products on UNIX Systems" on page 71.

Uninstall Products Using Software AG Installer

Uninstall EntireX

When shutting down products, also shut down all EntireX Brokers, RPC Servers, and customer applications that use EntireX libraries. For instructions, see the product documentation.

If you are going to uninstall from a UNIX system, and you used sudo to register daemons and set file permissions during or after installation, you must also use sudo to unregister daemons and remove file permissions, as follows:

- If you are not going to use an uninstallation script to uninstall, you can choose the uninstaller option to perform this task, or you can perform this task before starting the uninstaller by executing the script *Software AG_directory* /bin/beforeUninstallAsRoot.sh with root permission (that is, call the script using the root user account or the sudo utility.
- If you are going to use an uninstallation script, the uninstaller cannot execute the script because it does not store the sudo password, for security reasons. You must execute the script yourself before starting the uninstaller.

Uninstall All Other Products

Follow the instructions in *Using Software AG Installer*, with the additional guidelines below.

If you are going to uninstall Software AG Designer Eclipse, and you want to review or preserve Eclipse installation information, do so before starting the uninstaller, because the uninstaller will delete the entire Software AG Designer installation directory. User-created data that is stored in other directories, such as your workspace, will remain untouched.

If you are going to uninstall Integration Server or Microservices Runtime, retract all assets you have published to CentraSite. For instructions, see the *webMethods BPM and CAF CentraSite Metadata Help*.

If you are going to uninstall CentraSite:

- If the CentraSite Registry Repository is on a different machine than the Application Server Tier, uninstall the Application Server Tier first, and then uninstall the Registry Repository.
- The CentraSite Registry Repository database will not be deleted after uninstallation is complete. There are many circumstances in which you should retain the database after uninstalling CentraSite; for example, when the database contains information that is important for your enterprise. If you do want to delete the CentraSite database after uninstallation is complete, Software AG strongly recommends making a backup copy of its contents first. Then delete the *Software AG_directory* /CentraSite/data directory.

If you are going to uninstall from a Mac OS or other UNIX system, and you used sudo to register daemons and set file permissions during or after installation, you must also use sudo to unregister daemons and remove file permissions, as follows:

- If you are not going to use an uninstallation script to uninstall, you can choose the uninstaller option to perform this task, or you can perform this task before starting the uninstaller by executing the script *Software AG_directory* /bin/beforeUninstallAsRoot.sh with root permission (that is, call the script using the root user account or the sudo utility.
- If you are going to use an uninstallation script, the uninstaller cannot execute the script because it does not store the sudo password, for security reasons. You must execute the script yourself before starting the uninstaller.

Create Database Users and Storage for PostgreSQL

You cannot create storage creation for PostgreSQL using Database Component Configurator. To create the database user and storage for PostgreSQL, use the PgAdmin tool or PSQL prompt.

1. Create the database and schema:

   ```
   CREATE DATABASE WM_PGDB;
   CREATE SCHEMA WM_PGDB_schema;
   ```

2. Set the search path for the schema:

   ```
   SET search_path TO WM_PGDB_schema;
   ```

3. Create the database user and set the password:

   ```
   CREATE USER WM_PGDB_user;
   ALTER USER WM_PGDB_user WITH ENCRYPTED PASSWORD 'xyz';
   ```

4. Set up the database user's default schema using the search path for the user:

   ```
   ALTER USER WM_PGDB_user SET search_path = WM_PGDB_schema;
   ```

5. Grant permissions to the database user to connect to the database and to use and create objects in the schema:

   ```
   GRANT CONNECT       ON DATABASE WM_PGDB      TO WM_PGDB_user;
   GRANT USAGE, CREATE ON SCHEMA WM_PGDB_schema TO WM_PGDB_user;
   GRANT ALL           ON SCHEMA WM_PGDB_schema TO WM_PGDB_user;
   ```

6. Grant permissions in case the database user needs to create objects in the future:

   ```
   ALTER DEFAULT PRIVILEGES IN SCHEMA WM_PGDB_schema GRANT ALL ON TABLES    TO
   WM_PGDB_user;
   ALTER DEFAULT PRIVILEGES IN SCHEMA WM_PGDB_schema GRANT ALL ON SEQUENCES TO
   WM_PGDB_user;
   ALTER DEFAULT PRIVILEGES IN SCHEMA WM_PGDB_schema GRANT ALL ON FUNCTIONS TO
   WM_PGDB_user;
   ALTER DEFAULT PRIVILEGES IN SCHEMA WM_PGDB_schema GRANT ALL ON TYPES     TO
   WM_PGDB_user;
   ```

Create Database Users and Storage in Non-DBO Schemas for SQL Server

If you are going to create Software AG database components in a SQL Server RDBMS, you can create them in non-dbo schemas if necessary. To do so, use SQL Server Studio rather than the Database Component Configurator to create the database, database user, and storage, and then use the Database Component Configurator to create the database components.

1. Create the database using SQL Server Studio.

2. Connect to the master database and then create a login for the new database as follows:

   ```
   CREATE LOGIN login_name WITH PASSWORD = 'password'
   ```

3. Run this command to make the new database the default database:

   ```
   ALTER LOGIN login_name WITH DEFAULT_DATABASE = database_name
   ```

4. Connect to the database and then run the SQL statements below:

   ```
   CREATE USER user_name FOR LOGIN login_name
   CREATE SCHEMA schema_name
   ALTER AUTHORIZATION ON SCHEMA::schema_name TO user_name
   ALTER USER user_name WITH DEFAULT_SCHEMA=schema_name
   execute sp_addrolemember N'db_datareader', user_name
   execute sp_addrolemember N'db_datawriter', user_name
   execute sp_addrolemember N'db_ddladmin', user_name
   ```

Create Database Components Using Database Component Configurator

This section explains how to use the Database Component Configurator to create or drop database users and storage, and to create or drop Software AG database components. When creating these objects, follow the guidelines below.

- Never create your own database tables in the same schema or database as a Software AG database component.

- Software AG strongly recommends using the Database Component Configurator to create database storage, tablespaces, and filegroups, and to create database users for all RDBMSs, with two exceptions (see below). If you must instead create these objects manually (for example, because of security requirements), go to the *Software AG_directory* /common/db/DCC/Storage/scripts/*release*/*RDBMS* directory, open the *RDBMS*_str_c_ddl_user.sql file, and grant the permissions listed in that file.

  The two exceptions are PostgreSQL or when using non-DBO schemas for SQL Server. For these two cases, do not use the Database Component Configurator to create database users and storage, but instead follow the instructions in the previous sections.

- You can create one database user for multiple database components, or you can create a separate database user for each database component.

- You can create all database components in the same storage, or you can create each database component in its own storage.

- For Oracle and Tibero, Software AG scripts require ALTER SESSION permission at the time of creating database tables. You can revoke this permission after the tables have been created.

- The table below shows the storage convention that all Software AG database components follow by default. You can use a different convention when you create storage. (For PostgreSQL there are no storage conventions and tablespaces are not used.)

RDBMS	Data Storage Unit	Index Storage Unit	BLOB Storage Unit
DB2 tablespaces	WEBMDATA	WEBMINDX	WEBMBLOB
MySQL	Default	Default	Default
Oracle or Tibero tablespaces	WEBMDATA	WEBMINDX	WEBMDATA
SQL Server filegroups	Primary	Primary	Primary

Install the Database Component Configurator and Database Scripts

1. Download the Software AG Installer to any machine on the same network as your database server.

2. Start the Software AG Installer. In the product selection list, select **Database Configuration**, then continue through the installation process.

3. The Database Component Configurator has default environment settings. Go to the *Software AG_directory* \common\db\bin directory, open the setEnv.{bat|sh} file in the text editor, and edit the fields as necessary.

   The table below shows the default environment settings for Database Component Configurator.

Setting	Determines whether the configurator...	Default
-DtermOutOn	Writes execution information to the console.	true
-DspoolOn	Logs execution information.	true
-DlogLevel	Sets the log level to INFO (high-level information) or DEBUG (more detailed information, including internal processing and SQL calls).	INFO
-Dlog.dir	This setting specifies the full path to the directory in which to store the log files. Make sure you have permission to write to this directory.	*Software AG_directory* \common\db\logs

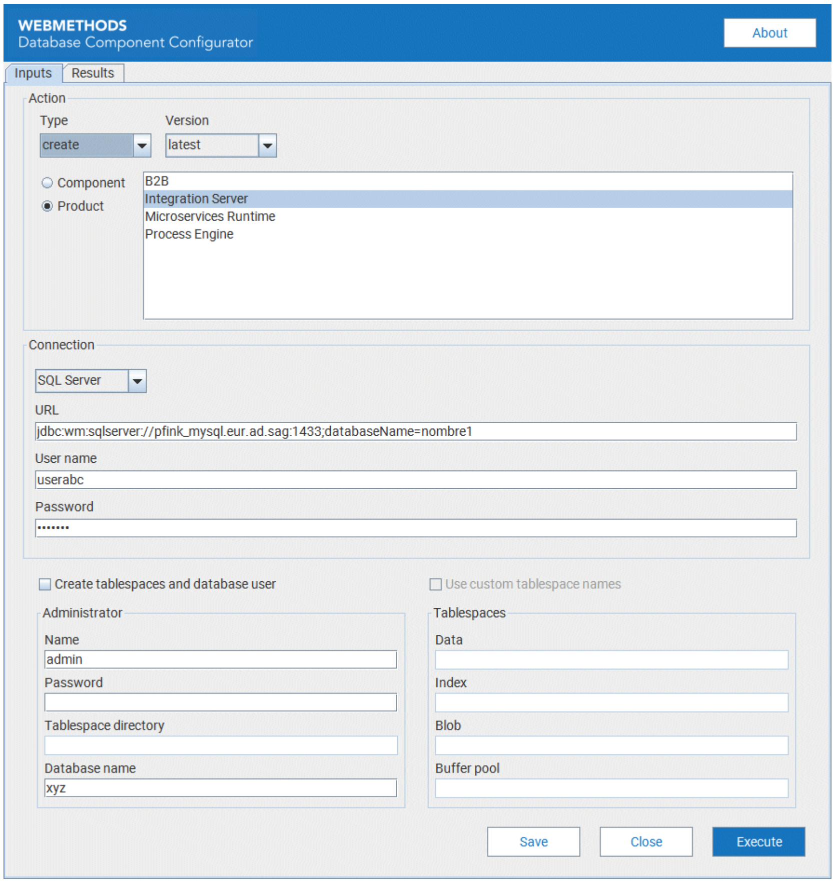
Database Component Configurator Graphical User Interface

You can use the Database Component Configurator GUI on all operating systems that are supported by your products. UNIX systems must have an X Windows environment. Each run of the configurator operates on a single schema.

The configurator writes execution information to the **Results** tab and to a log file named dcc.log in the *Software AG_directory* \common\db\logs directory. The maximum size of the file is 10MB. When that size is exceeded, the configurator creates a new file named dcc.*number*.log. If the number of files exceeds 1000, the configurator deletes the oldest file.

Start the Database Component Configurator GUI

System	Action
Windows	On the **Start** menu, go to **Programs > Software AG > Tools > Database Component Configurator**.
UNIX	Go to *Software AG_directory* and run the command `dbConfigurator.sh`.



Action

Type: create

Version: latest

○ Component

● Product

B2B
Integration Server
Microservices Runtime
Process Engine

Connection

SQL Server

URL: jdbc:wm:sqlserver://pfink_mysql.eur.ad.sag:1433;databaseName=nombre1

User name: userabc

Password: •••••••

□ Create tablespaces and database user

□ Use custom tablespace names

Administrator

Name: admin

Password:

Tablespace directory:

Database name: xyz

Tablespaces

Data:

Index:

Blob:

Buffer pool:

Initially, some of the fields on the **Inputs** tab show values you entered for the same fields in the Software AG Installer during product installation. You can change these values and specify other field values.

Create Database Components, Database User, and Storage

Choose the Action to Perform

In the **Action** area, in the **Type** list, click the action to perform, as follows:

Option	Action
create	Creates the database components you select in the **Action** area, and lets you create a database user and storage.
recreate	Drops and then re-creates the database components you select in the **Action** area.
catalog	List existing database components on the **Results** tab.

Create Database Components, Database User, and Storage

If you selected **create**, do the following:

1. In the **Action** area, do one of the following:

   - To create individual database components, click **Component** and select the database components.

   - To create all database components for products that have multiple database components, click **Product** and select the products. If you want to see the list of database components for a product, click **Product**, click **print** in the **Action Type** list, click **Execute**, and then click the Results tab.

2. For version, select **Latest**.

3. If no database user and storage already exist from a previous create action, follow the steps below.

   a. Select the check box labeled as follows:

RDBMS	Label
DB2	**Create Tablespaces and Grant Permissions to OS User**
MySQL	**Create Database and Database User**
Oracle or Tibero	**Create Tablespaces and Database User**
SQL Server	**Create Database and Database User**

For DB2 on Linux systems, tablespaces are created for each DB2 database. If you are creating database components in more than one DB2 database, either the tablespace directory or the tablespace names must be unique for each DB2 database.

b. In the **Admin ID** field, identify the database user or operating system user that has the necessary database administrator credentials. Supply the password for the user in the **Admin Password** field.

c. The next field and your entry depend on your RDBMS.

RDBMS	Field and Entry
DB2, Oracle, or Tibero	In the **Tablespace Directory** field, identify the directory in which to create the tablespaces.
MySQL or SQL Server	In the **Database** field, specify the database to create.

4. For DB2, Oracle, or Tibero, you can select the **Use Custom Tablespace Names** check box and specify custom tablespace names in the fields. Your products support all tablespace configurations deployed by users.

RDBMS	Entry and Notes
DB2	The custom tablespace names will replace the defaults WEBMDATA, WEBMINDX, and WEBMBLOB. You can also specify a custom name to use for the buffer pool for your products (WEBMBUFF by default).
Oracle or Tibero	The custom tablespace names will replace the defaults WEBMDATA and WEBMINDX.

Specify the Connection to the RDBMS

Specify the connection for the configurator to use to connect to the RDBMS.

1. Choose the RDBMS in which to create the database components and provide the URL for the RDBMS. Below is additional information for the URL.

   - For DB2, if you are going to create database components in a schema other than the default schema for the specified database user, you must specify these connection options in the URL:

     ```
     ;AlternateId=schema;"InitializationString=(SET CURRENT
     PATH=current_path,schema)"
     ```

     `AlternateID` is the name of the default schema used to qualify unqualified database objects in dynamically prepared SQL statements.

     If you are creating Optimize database components, you must specify *`schema`* using all uppercase letters. In addition, you must specify the options

`CreateDefaultPackage=true,ReplacePackage=true`, and `DynamicSections=3000`. These settings will affect all database components in the same schema or database.

If you are using a schema other than the default schema for the specified database user, you must also add the option `showSelectableTables=false` to the URL. The database driver queries SYSIBM.SYSDBAUTH at connection time to find out whether the current user is an admin or not. The results of this check allows a query to getTables() to return tables for which the user has SELECT privileges. An admin has implicit select privileges on all tables instead of the explicit privileges that other types of users have, so the driver has to alter its native SQL for getTables(). Set `showSelectableTables=false` and the driver will skip the check for SYSIBM.SYSDBAUTH at connection time.

- For Oracle, if you are creating storage and the Data Purge database component, you must specify the sysLoginRole connection option on the URL (for example, `;sysLoginRole=sysdba`).
- If you want to apply Kerberos functionality to set the current user to a specific name, set the connection URL to contain the parameter ReAuthenticateUser=*name*.
- For information about options supported by the DataDirect Connect JDBC driver used by your products, including options that support clustering, data encryption (SSL), and different authentication methods, see the DataDirect Connect documentation, available on the Software AG Documentation website.

2. The user ID and password you specify depend on the task you are going to perform, as described below.

Note:
You can choose to specify this authentication information in the URL, using DataDirect options, rather than in these fields.

If you are...	Specify...
Creating a database user and storage in DB2	OS user to which to grant permissions, and the password for that user.
Creating a database user and storage in Oracle, SQL Server, or Tibero	Database user and password to create. For SQL Server, the user will be created and a default schema named dbo will be assigned to that user. For Oracle or Tibero, do not use the SYSTEM user to create the database components in the SYSTEM schema. For Oracle 12c or later, do not create the database user in the Container Database (CDB). Instead, create the database use in the Pluggable Database (PDB), by specifying the PDB service name in the URL.
Performing any other action	Existing database user and password.

3. If you intend to run the configurator more than once, you can set the current field values as the defaults for subsequent runs by clicking **Save Settings**. The values for the two **Password** fields are not saved.

4. Click **Execute**.

Drop Database Components

1. Shut down all products that are connected to the database components you want to drop, and back up the database components.

2. In the **Action** area:

    a. In the **Type** list, click **catalog** to list existing database components on the **Results** tab. Then click **drop** as the action to perform.

    b. Select the individual database components to drop or the products whose database components to drop.

    c. For version, select **Latest**.

3. In the **Connection** area, click the RDBMS from which to drop the database components, type the URL for the RDBMS, and specify the database user and password to use to connect to the RDBMS.

4. Click **Execute**.

Drop the Storage and Revoke Database User Permissions

If you have dropped all database components in a schema, you can drop the storage and revoke the database user permissions.

1. In the **Action** area, in the **Type** list, click **drop**. Then click **Component** and select **All**.

2. In the **Connection** area, click the RDBMS from which to drop the database components, type the URL for the RDBMS, and specify the database user and password to use to connect to the RDBMS.

3. Select the **Drop tablespaces and database user** check box.

    a. In the **Admin ID** field, identify the database user or operating system user that has database administrator credentials to drop the database user and storage. Supply the password for the database user in the **Admin password** field.

    b. The next field depends on your RDBMS.

RDBMS	Fields and Entries
DB2, Oracle, or Tibero	In the **Tablespace directory** field, identify the directory that contains the tablespaces to drop. If you specified custom tablespace names, supply those names in the **Tablespaces** area.
MySQL or SQL Server	In the **Database** field, specify the database that contains the storage.

4. Click **Execute**.

Database Component Configurator Command

Go to the *Software AG_directory* \common\db\bin directory and run the command below.

Each run of the configurator operates on a single schema.

You must create database storage before you create database components.

Main Parameters

```
dbConfigurator.{bat|sh} {-a|--action} action
{-d|--dbms}{db2luw|mysql|oracle|sqlserver|pgsql|tibero}
{-l|--url} RDBMS_URL
{{-c|--component} db_component_name_or_code[,db_component_name_or_code...] |
  {-pr|--product} product_name} {-v|--version} latest
{-u|--user} db_user {-p|--password} password
[{-au|--admin_user} db_admin_user {-ap|--admin_password} password]
```

Additional Parameters

```
[{-tsdata|--tablespacefordata} data_tspace_name]
[{-tsindex|--tablespaceforindex} index_tspace_name]
[{-tsblob|--tablespaceforblob} BLOB_tspace_name]
[{-b|--bufferpool} buffer_pool_name]
[-t|--tablespacedir directory]
[-n|--dbname database_name]
[{-e|--export|-i|--import} {-dir|--configdir} directory  {-file|--configfile} file_name]

[-r|--runCatalog]
[-h|--help]
[-pa|--printActions] [-pc|--printComponents] [-pp|--printProducts]
[--resume]
```

Main Parameters

{-a | --action} ***action***

The table below describes the parameters for the {-a | --action} parameter.

Value	Action
`catalog`	Lists existing database components.

Value	Action
`create`	Creates a database user and database storage, or database components.
`drop`	Drops a database user and database storage, or database components. Before dropping any database component, shut down all products that are connected to the database component. After you drop all database components from a schema, you can drop the storage and revoke the database user permissions.
`recreate`	Performs the drop action and then the create action.

{-d | --dbms} {db2luw | mysql | oracle | sqlserver | pgsql | tibero}
Type of RDBMS on which to perform *`action`* .

{-l | --url} ***RDBMS_URL***
URL for the RDBMS. When working with database components on UNIX systems, you must enclose the URL in double quotes. The table below describes the parameters for the {-l | --url} parameter.

For information about options supported by the DataDirect Connect JDBC driver used by your products, see the DataDirect Connect documentation, available on the Software AG Documentation website.

RDBMS	Sample URL Format

DB2	`jdbc:wm:db2://server:{50000|port};databaseName=database [;option=value]...`
	For DB2, if you are going to create database components in a schema other than the default schema for the specified database user, you must specify these connection options in the URL:
	`;AlternateId=schema ;"InitializationString=(SET CURRENT PATH=current_path,schema)"`
	`AlternateID` is the name of the default schema used to qualify unqualified database objects in dynamically prepared SQL statements.
	If you are creating Optimize database components, you must specify *schema* using all uppercase letters. In addition, you must specify the options `CreateDefaultPackage=true`, `ReplacePackage=true`, and `DynamicSections=3000`. These settings will affect all database components in the same schema or database.
	If you are using a schema other than the default schema for the specified database user, you must also add the option `showSelectableTables=false` to the URL. The database driver queries SYSIBM.SYSDBAUTH at connection time to find out whether the current user is an admin or not. The results of this check allows a query to getTables() to return tables for which the user has SELECT privileges. An admin has implicit select privileges on all tables instead of the explicit privileges that other types of users have, so the driver

RDBMS	Sample URL Format
	has to alter its native SQL for getTables(). Set `showSelectableTables=false` and the driver will skip the check for SYSIBM.SYSDBAUTH at connection time.
MySQL Community Edition	`jdbc:mysql://`*`server`*`:{3306|`*`port`*`}/`*`database`*

MySQL Enterprise Edition	`jdbc:wm:mysql://`*`server`*`:{3306|`*`port`*`};databaseName=`*`database`*
	`[;`*`option`*`=`*`value`*`]...`

Oracle	`jdbc:wm:oracle://`*`server`*`:{1521|`*`port`*`};serviceName=`*`service`*
	`[;`*`option`*`=`*`value`*`]...`
	If you are creating storage and the Data Purge database component, you must specify the sysLoginRole connection option on the URL (for example, `;sysLoginRole=sysdba`).

PostgreSQL	`jdbc:wm:postgresql://`*`server`*`:{5432|`*`port`*`};databaseName=`*`database`*

SQL Server	`jdbc:wm:sqlserver://`*`server`*`:{1433|`*`port`*`};databaseName=`*`database`*
	`[;`*`option`*`=`*`value`*`]...`

Tibero	`jdbc:tibero:thin:@`*`server`*`:{8629|`*`port`*`}:`*`database`*

{-c|--component} ***db_component*****[,*****db_component...*****]** **|** **{-pr|--product}** ***product*****[,*****product...*****]}** **{-v|--version} latest**

One of the following:

- To create database storage, specify `storage` on `{-c|--component}`.
- To create individual database components, specify the database component codes or names on `{-c|--component}`. `--printComponents` provides a list of database component codes and names.
- To create all database components for products that have multiple database components, specify the product name or names on `{-pr|--product}`. `--printProducts` provides a list of these product names and codes.

{-u | --user} ***db_user*** **{-p | - -password}** ***password***

The table below describes the parameters for the {-u | --user} and {-p | - -password} parameters.

> **Note:**
> You can choose to specify this authentication information in the URL, using DataDirect options, rather than in these fields.

Action	Values to specify
Creating a database user and storage in	Database user and password to create.

Action	Values to specify
Oracle, SQL Server, or Tibero	For SQL Server, the user will be created and a default schema named dbo will be assigned to that user. For Oracle or Tibero, do not use the SYSTEM user to create the database components in the SYSTEM schema.
Creating a database user and storage in DB2	OS user to which to grant permissions, and the password for that user.
Performing any other action	Existing database user and password.

[{-au | --admin_user} *db_admin_user* **{-ap | --admin_password}** *password***]**
If no database user and storage already exist from a previous create action, or if you are going to drop storage and revoke the database user permissions, specify the database user or operating system user and password that has the necessary database administrator credentials.

Additional Parameters

[{-tsdata | --tablespacefordata} *data_tspace_name***] [{-tsindex | -- tablespaceforindex}** *index_tspace_name***] [{-tsblob | -- tablespaceforblob}** *BLOB_tspace_name***] [{-b | --bufferpool}** *buffer_pool_name***]**
If you are going to create database components in DB2, Oracle, or Tibero, you can specify custom tablespace names. Your products support all tablespace configurations deployed by users. For Oracle or Tibero, the custom tablespace names will replace the defaults WEBMDATA and WEBMINDX. For DB2, the custom tablespace names will replace the defaults WEBMDATA, WEBMINDX, and WEBMBLOB. You can also specify a custom name to use for the buffer pool (WEBMBUFF by default). For DB2, Oracle, or Tibero, if you are going to drop storage and revoke the database user permissions, provide the custom tablespace names.

[{-t | --tablespacedir} *directory***]**
If you are going to create a database user and storage, and want to create a tablespace directory for DB2, Oracle, or Tibero, full path to the directory. For DB2 on Linux systems, tablespaces are created for each DB2 database. If you are creating database components in more than one DB2 database, either the tablespace directory or the tablespace names must be unique for each DB2 database. If you are going to drop storage and revoke the database user permissions for Oracle or DB2, identify the directory that contains the tablespaces to drop.

[{-n | --dbname} *database_name***]**
If you are going to create a database user and storage in SQL Server, name of the target database. If you are going to drop storage and revoke the database user permissions in SQL Server, identify the database that contains the storage.

[{-e | --export} {-dir | --configdir} *directory* **{-file | --configfile}** *file_name***]**
Exports values for all required parameters except the password parameters to an .xml file.

[{-i | --import} {-dir | --configdir} *directory* **{-file | --configfile}** *file_name***]**

Imports values for all parameters from an .xml file to which you exported. Password parameters are not exported to the file, so you must specify them on the command.

[-r | --runCatalog]
Runs the catalog action at the end of every execution.

[-h | --help]
Lists command line interface help.

[-pa | --printActions]
Lists the actions you can perform, like create or drop.

[-pc | --printComponents]
Lists all database component names and codes.

[-pp | --printProducts]
Lists the names of products for which you can run {-pr|--product} *product*.

[--resume]
Database scripts are executed as sections, with each section concluding with a commit to the database. If an action such as create fails (for example, because of a network outage, an expired connection, or invalid credentials), the failed section will not have been committed, and therefore can safely be re-attempted. To do so, re-enter the failed command and add the --resume option. You must specify the same values for all parameters except the credential parameters, which can be different if necessary.

Examples

Important:
On UNIX systems, you must enclose the RDBMS URL in your commands in double quotes.

Create the Database User and Storage

This command creates the buffer pool and tablespaces (default names), and grants permissions for DB2:

```
dbConfigurator.bat -a create -d db2luw -c storage -v latest
-l jdbc:wm:db2://vmxpdb01:50000;databaseName=amol -u webmuser -au adminuser
-ap admin_password -t tablespace_dir
```

This command creates the buffer pool and tablespaces (custom names), and grants permissions for DB2:

```
dbConfigurator.bat -a create -d db2luw -c storage -v latest
-l jdbc:wm:db2://vmxpdb01:50000;databaseName=amol -u webmuser -au adminuser
-ap admin_password -t tablespace_dir -tsdata MYDATA -tsindex MYINDEX
-tsblob MYBLOB -b MYBUFF
```

This command creates the database user and the two tablespaces (default names) for Oracle:

```
dbConfigurator.bat -a create -d oracle -c storage -v latest
-l jdbc:wm:oracle://DBserver:1521;serviceName=myservicename -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir
```

This command creates the database user and the two tablespaces (custom names) for Oracle:

```
dbConfigurator.bat -a create -d oracle -c storage -v latest
-l jdbc:wm:oracle://DBserver:1521;serviceName=myservicename -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir -tsdata MYDATA
-tsindex MYINDEX
```

This command creates the database user and storage for SQL Server:

```
dbConfigurator.bat -a create -d sqlserver -c storage -v latest
-l jdbc:wm:sqlserver://DBserver:1433;databaseName=master -u webmuser -p w3bmpass
-au sa -ap sa_password -n webmdb
```

This command creates the database user and the two tablespaces (default names) for Tibero:

```
dbConfigurator.bat -a create -d tibero -c storage -v latest
-l jdbc:tibero:thin:@DBserver:8629:mydbname -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir
```

This command creates the database user and the two tablespaces (custom names) for Tibero:

```
dbConfigurator.bat -a create -d tibero -c storage -v latest
-l jdbc:tibero:thin:@DBServer:8629:mydbname -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir -tsdata MYDATA
-tsindex MYINDEX
```

Drop the Storage and Revoke Database User Permissions

If you drop all database components in a schema, you can drop the storage and revoke the database user permissions.

This command drops the buffer pool and tablespaces (default names), and revokes permissions for DB2:

```
dbConfigurator.bat -a drop -d db2luw -c storage -v latest
-l jdbc:wm:db2://vmxpdb01:50000;databaseName=amol -u webmuser -au adminuser
-ap admin_password -t tablespace_dir
```

This command drops the buffer pool and tablespaces (custom names), and revokes permissions for DB2:

```
dbConfigurator.bat -a drop -d db2luw -c storage -v latest
-l jdbc:wm:db2://vmxpdb01:50000;databaseName=amol -u webmuser -au adminuser
-ap admin_password -t tablespace_dir -tsdata MYDATA -tsindex MYINDEX
-tsblob MYBLOB -b MYBUFF
```

This command drops the two tablespaces (default names) for Oracle:

```
dbConfigurator.bat -a drop -d oracle -c storage -v latest
-l jdbc:wm:oracle://DBserver:1521;serviceName=myservicename -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir
```

This command drops the two tablespaces (custom names) for Oracle:

```
dbConfigurator.bat -a drop -d oracle -c storage -v latest
-l jdbc:wm:oracle://DBserver:1521;serviceName=myservicename -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir -tsdata MYDATA
-tsindex MYINDEX
```

This command drops the database for SQL Server:

```
dbConfigurator.bat -a drop -d sqlserver -c storage -v latest
-l jdbc:wm:sqlserver://DBserver:1433;databaseName=master -u webmuser
-p w3bmpass -au sa -ap sa_password -n webmdb
```

This command drops the two tablespaces (default names) for Tibero:

```
dbConfigurator.bat -a drop -d tibero -c storage -v latest
-l jdbc:tibero:thin:@DBServer:8629:mydbname -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir
```

This command drops the two tablespaces (custom names) for Tibero:

```
dbConfigurator.bat -a drop -d tibero -c storage -v latest
-l jdbc:tibero:thin:@DBServer:8629:mydbname -au adminuser
-ap adminpass -u webmuser -p w3bmpass -t tablespace_dir -tsdata MYDATA
-tsindex MYINDEX
```

Create Database Components

This command displays the database components that currently exist in DB2:

```
dbConfigurator.bat -a catalog -d db2luw
-l jdbc:wm:db2://DBserver:50000;databaseName=webmdb -u webmuser -p w3bmpass
```

This command creates the latest ProcessAudit and ProcessEngine database components in DB2 when creating in ALTSCHEMA rather than the default schema for the specified database user:

```
dbConfigurator.bat -a create -d db2luw -c processaudit,processengine -v latest
-l "jdbc:wm:db2://DBserver:50000;databaseName=webmdb
;AlternateId=ALTSCHEMA;InitializationString=\SET CURRENT
PATH=current_path,ALTSCHEMA\" -u webmuser -p w3bmpass
```

This command creates the latest versions of the ProcessAudit and ProcessEngine database components in Oracle:

```
dbConfigurator.bat -a create -d oracle -c processaudit,processengine -v latest
-l jdbc:wm:oracle://DBserver:1521;serviceName=myservicename
-u webmuser -p w3bmpass
```

This command creates the latest Trading Networks database components in PostgreSQL:

```
dbConfigurator.bat -a create -d pgsql -pr TN -v latest
-l jdbc:wm:postgresql://vmblrdb-4:5432;databaseName=dccpgdb
-u postgres -p manage
```

This command recreates (drops and then creates) the latest versions of the ProcessAudit and ProcessEngine database components in SQL Server:

```
dbConfigurator.bat -a recreate -d sqlserver -c processaudit,processengine -v latest
-l jdbc:wm:sqlserver://DBserver:1433;databaseName=webmdb -u webmuser -p w3bmpass
```

This command creates the latest versions of the ProcessAudit and ProcessEngine database components in Tibero:

```
dbConfigurator.bat -a create -d tibero -c processaudit,processengine -v latest
-l jdbc:tibero:thin:@DBServer:8629:mydbname
```

```
-u webmuser -p w3bmpass
```

Drop Database Components

Before dropping, shut down all products that are connected to the database component and back up the database component.

The command drops the latest version of the Analysis database component from DB2:

```
dbConfigurator.bat -a drop -d db2luw -c analysis -v latest
-l jdbc:wm:db2://DBserver:50000;databaseName=webmdb -u webmuser -p w3bmpass
```

This command drops the latest version of the Analysis database component from DB2 when dropping from ALTSCHEMA rather than the default schema for the specified database user:

```
dbConfigurator.bat -a drop -d db2luw -c analysis -v latest
-l jdbc:wm:db2://DBserver:50000;databaseName=webmdb;AlternateId=ALTSCHEMA
;"InitializationString=(SET CURRENT PATH=current_path,ALTSCHEMA)"
-u webmuser -p w3bmpass
```

This command drops the latest ProcessAudit and ProcessEngine database components from Oracle:

```
dbConfigurator -a drop -d oracle -c processaudit,processengine -v latest
-l jdbc:wm:oracle://myserver:1521;serviceName=myservicename -u webmuser
-p w3bmpass
```

This command drops the latest versions of all database components from SQL Server:

```
dbConfigurator.bat -a drop -d sqlserver -c core -v latest
-l jdbc:wm:sqlserver://myserver:1433;databaseName=webmdb -u webmuser -p w3bmpass
```

This command drops the latest ProcessAudit and ProcessEngine database components from Tibero:

```
dbConfigurator -a drop -d tibero -c processaudit,processengine -v latest
-l jjdbc:tibero:thin:@DBServer:8629:mydbname -u webmuser
-p w3bmpass
```

Pass Parameters to Database Component Configurator Using a Shell Script or Bat File

You can create a shell script or bat file that passes parameters, including passwords, to the Database Component Configurator.

This Linux shell script passes parameters to the Database Component Configurator:

```
% echo './bin/dbConfigurator.sh
--component "ISI" --action "drop" --version latest
--url "jdbc:wm:oracle://host:port;serviceName=name"
--user "user" --password "password"
--admin_user "admin_user" --admin_password "admin_password"
--dbms Oracle --dbname "name" --tablespacedir "c:\\app\\test"' > run-isi-drop.sh
% sh ./run-isi-drop.sh
```

This Windows bat file passes parameters to the Database Component Configurator:

```
c:\dir\>echo bin\dbConfigurator.bat
--component "ISI" --action "drop" --version latest
--url "jdbc:wm:oracle://host:port;serviceName=name"
```

```
--user "user" --password "password"
--admin_user "admin_user" --admin_password "admin_password"
--dbms Oracle --dbname "name" --tablespacedir "c:\\app\\test" > run-isi-drop.bat
c:\dir\>.\run-isi-drop.bat
```

Connecting Products to Database Components

This section explains how to connect Integration Server and hosted products to their database components. For information on connecting other products such as Optimize to their database components, see the product documentation.

You connect Integration Server to database components as follows:

- Define *JDBC database connection pools* that specify the connections to database servers that host database components.

- Direct predefined *functions* to write to their database components by pointing each function at the appropriate connection pool. For example, point the ISCoreAudit Log function to the connection pool for the ISCoreAudit database component, point the Xref function to the connection pool for the CrossReference database component, and so on. At run time, Integration Server creates a separate instance of the connection pool for each database component.

Integration Server can connect to the ActiveTransfer and ActiveTransferArchive, APIGatewayEvents, Archive, BusinessRules, CloudStreamsEvents, CrossReference, DocumentHistory, ISInternal, ISCoreAudit, MywebMethodsServer, MobileSupport, ProcessAudit, ProcessEngine, and TradingNetworks and TradingNetworksArchive database components.

For instructions on creating connection pools and pointing functions at them, see *webMethods Integration Server Administrator's Guide* and the points below.

- If you created the BusinessRules database component, you created it in the same schema or database as the ProcessAudit database component. Integration Server therefore writes to the BusinessRules database component using the ProcessAudit function. There is no separate function for the BusinessRules database component.

- If you installed the Integration Server and Optimize database components in two different schemas, you will have two instances of the Process Audit Log database component. Configure Integration Server and Optimize to point to the same instance. You can then drop the other, redundant instance.

A Registering Daemons to Automatically Start and Shut Down Products on UNIX Systems

Overview

Each Software AG daemon comes with an rc-script which can start and stop the daemon. For many products, you can register daemons at installation time. After installation, you can use the command line tool daemon.sh to register daemons for more Software AG products.

After successful daemon registration the operating system will call the rc-script with 'start' option at startup and with 'stop' option at shutdown. You can also use the daemon.sh tool to list the daemon-related files and to unregister daemons for Software AG products.

The daemon.sh tool supports the following types of UNIX daemons: initd (Solaris), inittab (AIX), launchd (macOS) and systemd (RHEL 7+, SLES 12+). While all these daemon types require different system commands for the (un-)registration of daemons, the daemon.sh works as a wrapper. You can use the daemon.sh tool to (un-)register daemons with a simple command line, which is the same on all supported UNIX operating systems. The following types of files are used:

- System file

  The daemon.sh script generates an init-script (initd, inittab), service file (systemd) or plist file (launchd) for each daemon registration and then it calls the individual system commands to complete that registration. The naming convention for the generated system file is `sag[`*`number`*`]`*`instance`* , where `sag` is a fixed prefix string, and *instance* is a unique identifier for the daemon instance. If you accidentally specify a non-unique *instance* during daemon registration, the daemon.sh script automatically adds *number* to make the init-script name unique. The system file is owned by the root user and the daemon (un-)registration requires root privileges.

  The table below shows the locations of system files generated by the daemon.sh script.

System	Location
Solaris	/etc/init.d
SLES 12, RHEL 7	/usr/lib/systemd/system
Mac OS	/Library/LaunchDaemons
AIX	/etc

- Start/stop shell script (rc-script)

  Each product daemon has an rc-script that has these features:

  - It is installed with the corresponding product.
  - It is owned and called by the installation user (that is, the non-root UNIX user that performed the installation).
  - It accepts the arguments 'start' and 'stop' to start and stop the corresponding product.
  - Some of these files are owned by a third party and might offer options 'install' or 'uninstall', which are not supported.

The rc-script names and locations are listed in "Product Daemons and rc-scripts" on page 77.

Note:
Products not listed in this appendix have no daemons to register or have their own method for registering daemons that is covered in product documentation.

Run the daemon.sh Command Line Tool

The daemon.sh command line tool is located in the *Software AG_directory*/common/bin directory. Log on to your system as the root user and call it from any current working directory.

The rc-script names and locations are listed in "Product Daemons and rc-scripts" on page 77.

Important:
The listed rc-scripts might offer options to install or uninstall, but these are daemon registration methods implemented by Tanuki Software, Ltd. and are not supported by Software AG. Use only the start or stop options with the rc-scripts, and use only daemon.sh to register or un-register daemons.

The table below lists the most commonly used command line options for the daemon.sh command line tool; they are the same for every supported UNIX platform.

Action	Command with Option
Register daemon with default system file name	daemon.sh -f *absolute_path_to_rc-script*
Register daemon and specify instance name	daemon.sh -f *absolute_path_to_rc-script* -n *instance*
List all registered daemons that have prefix sag	daemon.sh -L
Unregister daemon	daemon.sh -r -f *absolute_path_to_rc-script*
Remove daemon registrations for which rc-script no longer exists (clean up)	daemon.sh -C
Print information about implementation and other command line options	daemon.sh -h
For RHEL 7+ and SLES12+ (systemd), Register daemon to start before a dependent daemon	daemon.sh -f absolute_path_to_rc_script -B dependent_service_file_basename
For RHEL 7+ and SLES12+ (systemd), Register daemon to start after a dependent daemon	daemon.sh -f absolute_path_to_rc_script -A dependent_service_file_basename
For RHEL 7+ and SLES12+ (systemd), Load non-default login shell environment	daemon.sh -f absolute_path_to_rc_script -E comma_separated_ list_of_environment_files
For RHEL 7+ and SLES 12+ (systemd),Ignore any login shell environment	daemon.sh -f absolute_path_to_rcscript -E-

Note:

On SLES12+ or RHEL7+, the daemon.sh option '-f' will create and enable a systemd service. Your daemon will only be started/stopped correctly if the corresponding systemd service is in 'active' state. In order to turn a service into 'active' state, you need to reboot the system or start the service with the command 'systemctl start SERVICE', where SERVICE is the basename of the systemd service file. You can see all your systemd service files with the command 'daemon.sh -L' (left column).

Configure Integration Server Shutdown Timeouts in systemd and Tanuki Service Wrapper (RHEL 7+, SLES12+, systemd)

If the regular shutdown time of Integration Server exceeds the configured timeout values for the Tanuki Service Wrapper or systemd, then its daemon will be aborted (SIGKILL, kill -9). The same can happen for any other Software AG daemon. If you have identified such an issue, you need to stop the daemon and manually configure higher timeout values for Tanuki and systemd. The example below shows how to take these actions for Integration Server.

Note:
If you later re-register the daemon with daemon.sh, this manual change gets reset back to the default.

- For Tanuki, edit the custom_wrapper.conf file for the Integration Server instance, change the property for shutdown timeout to 500 seconds:

  ```
  $ vi /opt/softwareag/profiles/IS*/configuration/custom_wrapper.conf
     wrapper.shutdown.timeout=500
  ```

- For systemd, stop the service, identify and edit the service file, add a shutdown timeout of 500 seconds and activate the service by starting the daemon:

  ```
  # systemctl stop sagis1105
  # daemon.sh -L
      /usr/lib/systemd/system/sag2is1105.service  /opt/softwareag/profiles/
                                                   IS_default/bin/sagis105
      /usr/lib/systemd/system/sag2spm105.service /opt/softwareag/profiles/SPM/
                                                   bin/sagspm105
      /usr/lib/systemd/system/sagnserverdaemon.service  /opt/softwareag/
                         UniversalMessaging/server/umserver/bin/nserverdaemon
  # vi /usr/lib/systemd/system/sag2is1105.service
     TimeoutStopSec=500  <<< add/change this property in the [Service] section.
  # systemctl daemon-reload
  # systemctl start sagis1105
  ```

Daemon Registration Examples for SLES 12+ and RHEL 7+ (systemd)

Register a daemon for Universal Messaging (SLES 12+, RHEL 7+, systemd) and Specify Startup Order

Suppose you installed Integration Server and Universal Messaging and during installation you selected **Register Windows service/UNIX daemon for automatic startup** for Integration Server and its dependent component Platform Manager. To have a daemon registration for Universal Messaging you would perform these steps:

1. Set the PATH variable for convenient usage of daemon.sh:

```
# PATH=/opt/softwareag/common/bin:$PATH
```

2. Check the PATH variable:

```
# which daemon.sh
    /opt/softwareag/common/bin/daemon.sh
```

3. Register the daemon for Universal Messaging:

```
# daemon.sh -f /opt/softwareag/UniversalMessaging/server/umserver/
                        bin/nserverdaemon
    [daemon.sh] Created /usr/lib/systemd/system/sagnserverdaemon.service(ok)
    [daemon.sh] chmod 644 /usr/lib/systemd/system/sagnserverdaemon.service(ok)
    Created symlink from /etc/systemd/system/multi-user.target.wants/
                                          sagnserverdaemon.service to
    /usr/lib/systemd/system/sagnserverdaemon.service.
    [daemon.sh] systemctl enable sagnserverdaemon.service (ok)
```

4. Identify the systemd service file names from the left column:

```
# daemon.sh -L
    /usr/lib/systemd/system/sag2is1101.service /opt/softwareag/profiles/
                                               IS_default/bin/sagis105
    /usr/lib/systemd/system/sag2spm101.service /opt/softwareag/profiles/SPM/
                                               bin/sagspm105
    /usr/lib/systemd/system/sagnserverdaemon.service /opt/softwareag/
                        UniversalMessaging/server/umserver/bin/nserverdaemon
```

5. Check if the status of the Universal Messaging daemon is **inactive (dead)**:

```
# systemctl status sagnserverdaemon
    ● sagnserverdaemon.service - sagnserverdaemon
       Loaded: loaded (/usr/lib/systemd/system/sagnserverdaemon.service;
                                        enabled; vendor preset: disabled)
       Active: inactive (dead)
```

6. Reboot the system or activate all the new systemd services:

```
# systemctl start sag2spm105
# systemctl start sag2is1105
# systemctl start sagnserverdaemon
```

7. Check if the status of the Universal Messaging daemon is **active**:

```
# systemctl status sagnserverdaemon
    ● sagnserverdaemon.service - sagnserverdaemon
       Loaded: loaded (/usr/lib/systemd/system/sagnserverdaemon.service;
                                       enabled; vendor preset: disabled)
       Active: active (running) since Fri 2019-03-08 13:22:18 CET; 12s ago
      Process: 11439 ExecStart=/bin/su - sag -c /opt/softwareag/
                UniversalMessaging/server/umserver/bin/nserverdaemon start
     (code=exited, status=0/SUCCESS)
    Mar 08 13:22:06 sles12 systemd[1]: Starting sagnserverdaemon...
    Mar 08 13:22:06 sles12 su[11439]: (to sag) root on none
    Mar 08 13:22:06 sles12 su[11439]: pam_unix(su-l:session): session opened
                                                      for user sag by (uid=0)
    Mar 08 13:22:18 sles12 systemd[1]: Started sagnserverdaemon.
```

Suppose you want Universal Messaging to start before Integration Server at system boot time. You would perform these steps:

1. Identify the service files and the rc-scripts for Integration Server and Universal Messaging:

```
# daemon.sh -L
    /usr/lib/systemd/system/sag2is1105.service  /opt/softwareag/profiles/
                                                 IS_default/bin/sagis105
    /usr/lib/systemd/system/sag2spm105.service /opt/softwareag/profiles/SPM/
                                                 bin/sagspm105
    /usr/lib/systemd/system/sagnserverdaemon.service  /opt/softwareag/
                        UniversalMessaging/server/umserver/bin/nserverdaemon
```

2. Register the daemon for Integration Server and add the `After=sagnserverdaemon.service` property to the service definition:

```
# daemon.sh -f /opt/softwareag/profiles/IS_default/bin/sagis105
                       -A sagnserverdaemon.service
```

3. Register again the daemon for Universal Messaging and add the `Before=sag2is1105.service` property to the service definition:

```
# daemon.sh -f /opt/softwareag/UniversalMessaging/server/umserver/bin/
                                           nserverdaemon -B sag2is1105.service
```

4. Activate both systemd services:

```
# systemctl start sag2is1105
# systemctl start sagnserverdaemon
```

Register Daemon for Universal Messaging (Solaris, initd) and Specify Startup Order

Suppose you installed Integration Server and Universal Messaging and during installation you selected **Register Windows service/UNIX daemon for automatic startup** for Integration Server and its dependent component Platform Manager. To have a daemon registration for Universal Messaging you would perform these steps:

1. Set the PATH variable for faster usage of daemon.sh:

```
# PATH=/opt/softwareag/common/bin:$PATH
```

2. Check the PATH variable:

```
# which daemon.sh
    /opt/softwareag/common/bin/daemon.sh
```

3. Register the daemon for Universal Messaging:

```
# daemon.sh -f /opt/softwareag/UniversalMessaging/server/umserver/bin/
                        nserverdaemon
    [daemon.sh] Created /etc/init.d/sagnserverdaemon (ok)
    [daemon.sh] chmod 775 /etc/init.d/sagnserverdaemon (ok)
    [daemon.sh] ln -s /etc/init.d/sagnserverdaemon /etc/rc3.d/
```

```
                       S20sagnserverdaemon (ok)
    [daemon.sh] chmod 775 /etc/rc3.d/S20sagnserverdaemon (ok)
    [daemon.sh] ln -s /etc/init.d/sagnserverdaemon /etc/rc2.d/
                       K20sagnserverdaemon (ok)
    [daemon.sh] chmod 775 /etc/rc2.d/K20sagnserverdaemon (ok)
```

4. List all registered daemons:

```
# daemon.sh -L
/etc/init.d/sag4is1105   /opt/softwareag/profiles/IS_default/bin/sagis105
/etc/init.d/sag4spm105   /opt/softwareag/profiles/SPM/bin/sagspm105
/etc/init.d/sagnserverdaemon /opt/softwareag/UniversalMessaging/server/
                              umserver/bin/nserverdaemon
```

After installing the daemons, it is still undefined whether Universal Messaging will start before or after Integration Server at system boot time.

The sequence for the daemon startup depends on an order number, where each daemon gets assigned a particular start number. If you have two daemons, then the daemon with the lower number will start first. You can specify the daemon start number with the daemon.sh option '-S'. The default in daemon.sh is '-S 20'. If you do not specify any number then all get the same default and the sequence is undefined. If you want Universal Messaging to start before Integration Server, specify a lower start number like '-S 10'.

A similar handling applies for the shutdown sequence, where we have a stop (kill) number that can be specified with the daemon.sh option '-K'. The following example shows how to change the start number and the kill number from default (20) to 10 for the Universal Messaging daemon, in order to make it start before and stop after Integration Server.

```
# daemon.sh -f /opt/softwareag/UniversalMessaging/server/umserver/bin/
                        nserverdaemon -S 10 -K 10
[daemon.sh] /etc/rc2.d/K20sagnserverdaemon removed (ok)
[daemon.sh] /etc/rc3.d/S20sagnserverdaemon removed (ok)
[daemon.sh] /etc/init.d/sagnserverdaemon removed (ok)
[daemon.sh] Created /etc/init.d/sagnserverdaemon (ok)
[daemon.sh] chmod 775 /etc/init.d/sagnserverdaemon (ok)
[daemon.sh] ln -s /etc/init.d/sagnserverdaemon /etc/rc3.d/S10sagnserverdaemon(ok)
[daemon.sh] chmod 775 /etc/rc3.d/S10sagnserverdaemon (ok)
[daemon.sh] ln -s /etc/init.d/sagnserverdaemon /etc/rc2.d/K10sagnserverdaemon(ok)
[daemon.sh] chmod 775 /etc/rc2.d/K20sagnserverdaemon (ok)
```

The command output shows all the file operations that go along with the initd way of daemon registration. Note that the previous specification gets automatically overwritten (removed). You can see that the new start and kill numbers become part of the symbolical link names, like 'S10' in S10sagnserverdamon or 'K10' in K10sagnserverdaemon.

Product Daemons and rc-scripts

When using the `daemon.sh -f` option to register or unregister a daemon, make sure to use the absolute path name for the rc-script. Below is an example that applies to most products.

```
daemon.sh -f /opt/softwareag/API_Portal/server/sagyaprelease.sh
```

For products which can have multiple instances, such as Universal Messaging, Integration Server, and My webMethods Server, the command has an additional option, `-n` *`instance`*. The command to use for Universal Messaging, for example, is as follows ....

```
daemon.sh -n umserver -f /opt/softwareag/UniversalMessasging/server/
umserver/bin/nserverdaemon
```

The table below shows the rc-script for each product daemon.

Product	**Daemon**	**rc-script (under *Software AG_directory*)**
ActiveTransfer Agent	Software AG ActiveTransfer Agent	profiles/MAG/bin/sagmag*release*
API Portal	APICloudAgent*release*	API_Portal/server/bin/CloudAgentApp.sh
CentraSite Registry Repository	CentraSite Registry/Repository	CentraSite/bin/saginm
EntireX Broker	EntireX Administration Service	EntireX/bin/sagexx
Integration Server	Software AG Integration Server	profiles/*instance*/bin/sagis*release*
My webMethods Server	Software AG My webMethods Server	profiles/*instance*/bin/sagmws*release_ instance_name_instance_number*
webMethods OneData	Software AG OneData Server	profiles/ODE/bin/sagode*release*
Optimize Infrastructure Data Collector	Software AG Optimize Infrastructure Data Collector	profiles/InfraDC/bin/sagwid*release*
Optimize Web Service Data Collector	Software AG Optimize WS Data Collector	optimize/dataCollector/bin/sagoptiDC.sh
Optimize Analytic Engine	webMethods Optimize Analysis Engine	optimize/analysis/bin/sagoptiAE.sh
Platform Manager	Software AG Platform Manager	profiles/SPM/bin/sagspm*release*
Software AG Runtime	Software AG Runtime	profiles/CTP/bin/sagctp*release*
Universal Messaging	nserverdaemon	UniversalMessaging/server/*instance*/bin/nserverdaemon

Troubleshooting

On AIX, the maximum length for an init-script name is limited to 14 characters (see the man-page for the mkitab command). If daemon.sh reports an error with the mkitab command, you can specify a shorter init-script name with the -n option of daemon.sh. Using `-n` *`instance`* is just a recommendation; you can use any shorter name.

On SLES and RHEL, you can use the following commands to view the status of a systemd service:

- `# systemctl status SERVICE` (where SERVICE is the basename of the systemd service file)
- `# journalctl -u SERVICE`

B Operating in International Operating Environments

Overview

Software AG products are designed for use anywhere in the world, and enable you to exchange data with systems and users who speak languages other than your own.

The products use the Universal Character Set (ISO/IEC 10646-2, or Unicode) as the character set for all internal operations. At minimum, the products support the requirements of Unicode 3.0. HTML content, including administration tools, uses Unicode UTF-8 encoding.

For specific information on the testing, configuration, capabilities, and limitations of any product, see the product's readme.

Language Packs

The U.S. English files necessary to run your products is always installed with the products. However, for many products, you can also install language packs that provide translated user interfaces, messages, help, and documentation.

Most products support multiple languages at a time; you can install as many language packs as you need. For products that support multiple languages, your client setup (that is, browser language preference, default locale in which the product is started, and locale selected in your user profile) will determine which language you see. If you operate a product in a locale for which language packs are not available or installed, you will see U.S. English.

Language packs are available on all supported operating systems except Mac OS X.

Language Packs for Software AG Designer

By default, the language that Software AG Designer uses depends on your JRE default locale, which in turn depends on your operating system locale setting. If the operating system locale is not set to the language you want to use, you can override the setting for Software AG Designer by starting Software AG Designer with the -nl option. For example, open a command window and enter `eclipse.exe -nl ja_JP`.

Software AG Designer is based on the Eclipse platform and its projects. Software AG language packs do not include language packs for plug-ins provided by Eclipse projects. If you need language packs for those plug-ins, follow the steps below. For background information about the available translations, their completeness, and more, go to the Eclipse web site.

1. Start Software AG Designer.
2. Go to **Help > Install New Software**, click the **Available Software** tab, and then click **Add...**.
3. In the **Location** field, type the URL for Eclipse 4.19-compatible language packs and click **OK**.
4. Click **Refresh**, select the language pack for the language you need, and click **Install**. Confirm the language pack to install and click **Next**.

5. Read the license agreement. If you accept the terms, select the check box and click **Finish**.

6. Restart Software AG Designer.

Extended Character Set

Software AG offers an extended character set that extends Java's java.nio.Charset class to support additional installable character encodings and adds more than 870 custom character encodings. The extended character set allows you to use custom character encodings such as EBCDIC and OEM character encodings with Integration Server, adapters, and services.

Configure Browsers and JRE Font

For supported browser releases, see *System Requirements for Software AG Products*.

Some HTML interfaces for Software AG products detect the language setting for your browser. If you have installed language packs on your products, those interfaces try to supply the user interface, messages, and data in that language.

The table below explains how to indicate the languages to display in your browser for Software AG products.

Browser	Action
Firefox	Go to **Tools > Options > Content**, click **Choose**, and add the language you want to use to the top of the list.
Google Chrome	Go to **Settings > Show Advanced Settings > Languages**, click spell-checker settings, and add the language you want to use to the top of the list.
Internet Explorer	Go to **Tools > Internet Options**, click **Languages**, and add the language you want to use to the top of the list.
Safari	Go to **System Preferences > Language & Text > Language**, and add the language you want to use to the top of the list.

Your JRE might require modification to display non-ASCII data. If you see hollow boxes or black squares where you would expect to see non-ASCII characters, you might need to modify your JRE's font configuration file to provide an appropriate font for displaying these characters. For example, you might install a full Unicode font, such as Arial Unicode MS, or you can install a font for a particular language.

Configure the Proper Locale

The table below provides instructions on how to configure a locale if you are going to install or run your products in a locale other than U.S. English.

System	Action
Windows	These instructions are for Windows 7; instructions for other Windows systems vary slightly. 1. On the **Start** menu, go to **Control Panel > Regional and Language**. 2. Click the **Keyboards and Languages** tab. If the language you want to use is not yet installed, click **Install/Uninstall Languages** and install the language. In the **Display Language** area, select the language and click **OK**. 3. Make the display locale setting take effect by logging off Windows and then logging in again.
UNIX	Set your shell to the proper locale by setting the LANG environment variable. Run the appropriate command below: ■ In an sh shell: `LANG=`*`locale`*`; export LANG` ■ In a csh shell: `setenv LANG `*`locale`*