MICRO FOCUS

Atlas Planning and Tracking Suite 4.2

Installation and Configuration Guide

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK
http://www.microfocus.com

2017-04-18

Contents

About Atlas

Atlas is a web-based, lightweight Requirements tool focused on enabling business analysts, product managers, and other Project stakeholders to:

- Easily gather and collaborate their ideas using their favorite media (diagrams, work-flows, pictures, presentations, documents, videos).
- Organize and structure those ideas into well formed requirements.
- Prioritize and plan those requirements into a specific time-frame given their agile teams velocity.
- Track the progress of their agile teams work towards their requirements, independently of their agile tool of choice.

In addition to the *Online Help*, *Installation Guide*, and *Release Notes*, for more information about Micro Focus Atlas, refer to *Microfocus.com* and the *Microfocus Community*.

About Rhythm

Rhythm is an agile project tracking tool designed to allow you to:

- Organize, prioritize, and manage your Agile teams' backlogs.
- Plan your sprints, task out the work, and then track progress throughout the sprint.
- Get comprehensive visibility of all your Agile assets.

In addition to the Online Help, Installation Guide, and Release Notes, for more information about Rhythm, refer to *Microfocus.com* and the *Micro Focus Community*. The latest version of the Release Notes and all documentation can be found at *SupportLine*.

System Requirements

Atlas Planning and Tracking Suite has the following system requirements:

Server

Operating Systems	• Microsoft Windows Server 2012 R2 (64-bit) • Microsoft Windows Server 2012 (64-bit). • Microsoft Windows Server 2008 R2 SP2 (64-bit).
Hardware	Minimum 64-bit quad core system with 16 GB RAM.

Client

Browsers	• Internet Explorer 10+. • Mozilla Firefox, recent versions. • Google Chrome, recent versions. **Note:** Other products used with Atlas Planning and Tracking Suite, for example, Micro Focus Connect, have their own set of system requirements. Please refer to the *Release Notes* for those products.
Screen Resolution	Minimum 1680x1050 resolution.

Databases

Atlas Hub is a server application installed with Micro Focus Atlas. It contains the database that Micro Focus Atlas uses. Atlas Hub supports both 32-bit and 64-bit databases. The following databases have been tested and are supported:

- Microsoft SQL Server 2014
- Microsoft SQL Server 2012 Express
- Microsoft SQL Server 2012 SP1
- Microsoft SQL Server 2008 Express R2
- Microsoft SQL Server 2008 Express
- Microsoft SQL Server 2008 R2 SP2
- Microsoft SQL Server 2008 SP3
- Oracle Database 12c version 12.1.0.2.0
- Oracle Database 11g R2
- PostgreSQL 9.3

Important: Never modify database contents or vault files other than through the Atlas Hub client or the Server Administration Tool. Direct database manipulation is unsupported.

Database User and Password

The PostgreSQL database server installed by default with Atlas Hub has two default users created:

- Admin user = `postgres`

- Superuser = `Borland_Login`

The password for both is `Borland_123`.

When you use the default PostgreSQL database, you will be asked for a **System Password**. It is the same as the Admin user password, `Borland_123`.

Installation

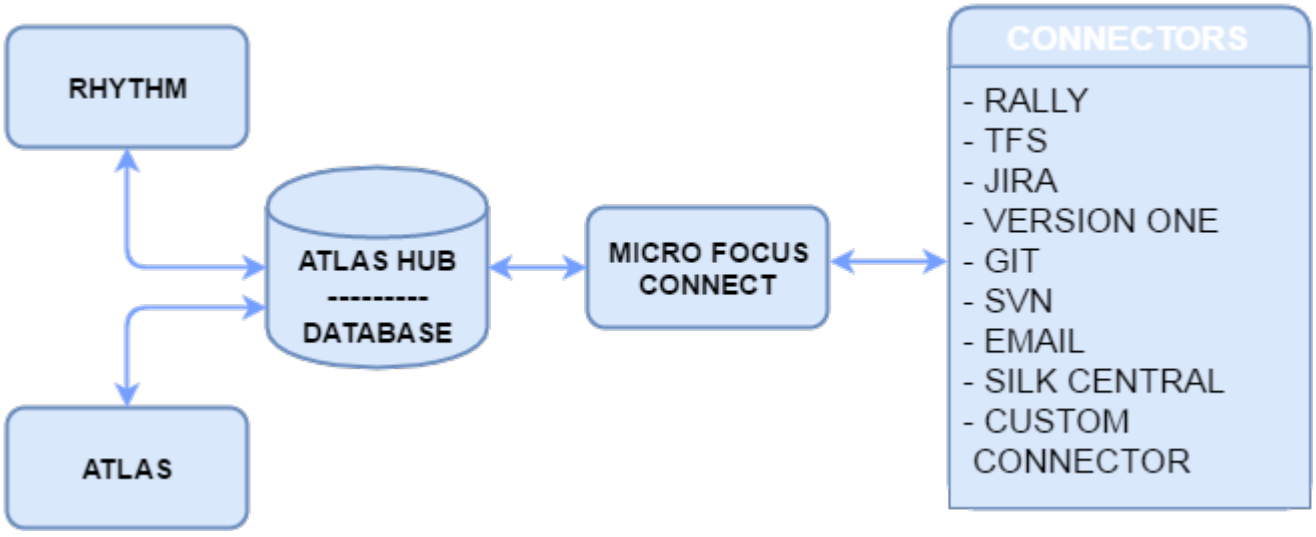
You are about to install the **Atlas Planning and Tracking Suite**. Before proceeding with installation, prepare the following items:

- A 64-bit quad core system with 16 GB RAM that meets the full set of system requirements.
- A database. If you do not have a database to use, a PostgreSQL database will be installed and configured by default. If you have one of the supported database types, you will need to continue to configure your database post installation.

A fully installed and configured **Atlas Planning And Tracking Suite** contains the following applications and components:

Atlas	A collaborative, flexible, agile requirements and delivery platform. It will be installed and run in a Tomcat server.
Rhythm	Rhythm is an agile project tracking tool designed to allow you to organize, prioritize, and manage your Agile teams' backlogs. You can plan your sprints, task out the work, and then track progress throughout the sprint. It provides comprehensive visibility of all your Agile assets. Rhythm should be used in the place of StarTeam Agile.
Atlas Hub	A software change and configuration management server that stores the assets.
Database	Atlas Hub connects to and supports many databases. If you do not have one installed, PostgreSQL will be installed.
Micro Focus Connect	A web application that allows you to synchronize assets from third party tools into Atlas Hub (and thereby, into Atlas). For example, you could connect Atlas Hub to Rally via Micro Focus Connect and have all stories from Rally appear in Atlas.
Tomcat Web Server	Two Tomcat web servers are installed. One runs Atlas, Rhythm, and Micro Focus Connect and the other runs the Search component.

The following depicts how the components are connected:



Installing the Atlas Planning and Tracking Suite

The following steps describe how to install the **Atlas Planning and Tracking Suite**. The suite includes Atlas, Micro Focus Connect, Rhythm, and Atlas Hub. These applications will all be installed together.

Please note all of the following before continuing with your installation:

- If you want to install the Atlas Hub on a separate server, you should plan to do that first. Go to that server, run the installer and select the **Atlas Hub** option on the **Choose Install Set** screen. Complete that installation. Then, on the server for the web applications, choose the **Atlas Planning and Tracking Suite** option, during the installation, you will be asked to point to an existing Atlas Hub, the one that you just set up.
- The base installation no longer configures StarTeam Agile to run. If you want to continue to use StarTeam Agile, you must remove the `_deprecated` from the `.WAR_deprecated` StarTeam Agile file.
- Unless you are upgrading from Atlas Planning and Tracking Suite 2.0, if you have any other version of Atlas, Rhythm, Micro Focus Connect, or the Atlas Hub installed, you must uninstall them before proceeding. If you are a current Micro Focus Connect customer and you want to maintain your mappings, you must retain a copy of `Connect.xml` before uninstalling. Afer installation, you can check your original `Connect.xml` file back into the `StarFlow Extensions/Connect` folder.
- We recommend backing up the Atlas Hub installation directory in order to preserve your configuration files.
- When upgrading from StarTeam Agile to Rhythm, it is necessary to migrate **Target Releases** from the StarTeam Agile Release Views to the Rhythm Releases. This migration is accomplished by running the below command using the StarTeam 16.x SDK that is installed with the Atlas Hub:

  ```
  stcmd rhythm-conversion -p "user:password@host:port/project name/"
  ```

  Running this command will do the following:

  - Create new Releases in Rhythm that match the **Target Releases** in StarTeam Agile. All Stories and Sprints that have a **Release** value set will also be updated. If you run the migration command after assigning **Release** values in Rhythm, then those assignments will be removed as the purpose of the migration is to match up the **Target Release** data in StarTeam Agile with Rhythm.
  - Each Story in the Project will receive updated values on the **Rank** and **Order** fields which are compatible with the Rhythm UI. The overall **Rank** and **Order** of the Stories in the Project **Backlog** will be retained and all new Story revisions will have a "Rhythm Conversion" comment.

1. Run the installer.

   Note: During an upgrade from a previous version, the Atlas Planning and Tracking Suite installer expects valid license keys. The Atlas Hub will fail to start if license keys are invalid.
2. Read the introduction and click **Next** on the **Introduction** screen.
3. Read the license agreement, select the option to accept the license agreement and click **Next**. The **Choose Install Set** screen appears. This is where you will be able to install either all of the applications or just install a new Atlas Hub.
4. To install all of the web applications, choose the **Atlas Planning And Tracking Suite** option to install Atlas, Rhythm, Micro Focus Connect, and optionally, Atlas Hub. Click **Next**.

   a) Select where you want the files installed on the **Choose Install Folder** screen. Do not click **Next** until you read the following:

   Important: The next part of the installation is where you choose the location. If you are upgrading, you need to make sure that you select the existing directory where the previous version was installed. If you used the default, you don't need to do anything in the next step because it is selected for you. However, if you chose a different location, you must browse to that folder and select it for the upgrade.

   If you are upgrading, you will receive an **Upgrade Required** message. Click **OK** and walk through the upgrade screens. Your upgrade will complete, and you are finished with the installation!

   If you are not upgrading, click **Next**. The **Atlas Hub Connection** screen appears.

   To connect to an existing Atlas Hub:

   1. Select **Connect to existing Atlas Hub**.

2. Click **Next**. The **Atlas Hub Parameters** screen appears.
3. Enter the Atlas Hub connection information:

Atlas Hub Host Name or IP Address	The IP address of an existing Atlas Hub installation. **Important:** When specifying the address and port for Atlas Hub in either the installation wizard or directly in the `ALMConfiguration.xml` file, you must use the actual IP address (or machine name) of the Atlas Hub. Do not use "`localhost`" even when Micro Focus Connect and the Atlas Hub are running on the same machine.
Configuration Name	The configuration name of an existing Atlas Hub installation.
Configuration GUID	The configuration GUID of an existing Atlas Hub installation.
Configuration Port	The configuration port of an existing Atlas Hub installation.
User Name	The user name of an existing Atlas Hub installation.
Password	The user's password of an existing Atlas Hub installation.

4. Click **Next**.

To install a new Atlas Hub:

1. Select **Install new Atlas Hub**.
2. Click **Next**. The **New Atlas Hub** screen opens.
3. Enter a **Configuration Name** for the new Atlas Hub.
4. If you already have a license key, select the **I have a license key** option.
5. Enter the license details.
6. Click **Next**.

5. To install the Atlas Hub only, choose the **Atlas Hub** option on the **Choose Install Set** screen.

Tip: After installation, if something goes wrong or a file is damaged or corrupted, you can attempt to repair the install by running this installer again

If you are upgrading, you need to:

1. Stop all Atlas Hub server configurations using the Atlas Hub **Admin Tool**.
2. Close the Atlas Hub **Admin Tool**.
3. On the installer screen, click **Next**.
4. Click **Install**. The installer will install all upgrade files.

You are finished upgrading the Atlas Hub!

If you are not upgrading, click **Next**.

1. Select the **Message Broker** option to install the Message Broker.
2. Select the **PostgreSQL** option to install a PostgreSQL database.
3. Click **Next**.
4. If you are installing a message broker, the **Message Broker Options** screen opens. Enter the port number on which the broker will listen.
5. If you are installing PostgreSQL, the **PostgreSQL Configuration** screen opens. Enter the password for the database superuser. Note that the username is `postgres`.
6. Click **Next**.

The **Choose Shortcut Folder** screen opens.

6. Select the shortcut folders and click **Next**. The Pre-Installation Summary screen opens.
7. Review the information and click **Install**.

Additional Configuration for Connecting to Existing Atlas Hub

If you chose the option to **Connect to existing Atlas Hub** in the Atlas installation wizard, the following configuration changes are required before using Micro Focus Connect with Atlas:

1. Open the Micro Focus Connect UI.
2. Navigate to the `HubDefault` data source.
3. Replace the default credentials with the correct account details for the Micro Focus Connect user (this should be a server administrator account) in the **User Name** and **Password** fields.
4. In the **URL** field, replace the default StarTeam URL with the correct URL value.
5. Save the data source.
6. Click (**Settings**) on the main Micro Focus Connect UI.
   a) Verify that the **Synchronization Frequency** field contains a valid integer.
   b) Verify that the **Maximum Number of Threads** field contains a valid integer.
   c) Click the **Error Notification** tab.
   d) In the **Email Settings** group, if there is a value in the **Frequency** field, verify that it contains a valid integer.
   e) Save the settings.
7. Go to the **Connections** page. For each connection, click (**Edit**). Look in the **Error Notification** group. If there is a value in the **Frequency** field, verify that it contains a valid integer.
8. Save the connection.
9. Restart the server.

Configuration

This section describes how to configure Atlas after installation. After the installation, you will need to consider things such as:

- Setting up a database other than the default PostgreSQL.
- Using Micro Focus Connect to synchronize Atlas with your other systems.
- Setting up email notifications for Atlas users.
- Changing application port numbers, if required.

Database Configuration

Atlas uses Atlas Hub to store its data. Therefore, you need to use the Server Administration Tool in Atlas Hub to configure your database.

Configuring Microsoft SQL Server/Express Databases

This chapter explains how to create a new server configuration to use Microsoft SQL Server or Microsoft SQL Server Express. Atlas Hub stores everything except for file archives and server startup information in the database of your choice.

You can use the Microsoft Windows version of Atlas Hub with the supported Microsoft databases listed in *Supported Databases*.

This chapter also provides an overview of the tuning and maintenance SQL scripts that are provided with Atlas Hub and explains how to use them. For detailed information on Microsoft SQL Server performance tuning, consult your Microsoft SQL Server documentation.

Terminology

When this guide uses the terms "instance" and "database", it uses Microsoft terminology. When you install Microsoft SQL Server on a computer, you can install up to 16 instances of it. Each instance can manage a number of different databases. Each Atlas Hub configuration uses its own database. When you perform a typical installation of the Atlas Hub, you install one instance of Microsoft SQL Server Express.

Logging Onto Databases

It is highly recommended that you use a dedicated user account to run or log onto the databases used with Atlas Hub configurations. System administrator accounts usually have unlimited privileges. Any anomalies or errors that occur while you are logged in as the system administrator may result in unrecoverable damage to databases and other databases managed by the same database server.

Understanding the Encoding Differences

Atlas Hub sends data encoded as UTF-8. Microsoft SQL Server and Microsoft SQL Server Express do not support UTF-8 at the database level. They support nchar, nvarchar, and ntext to store fixed format Unicode data (UTF-16).

- UTF-8 is a variable length character set in which the characters can expand from one to six bytes depending on the language.
- UTF-16 is a fixed length encoding mechanism in which every character expands to two bytes. UTF-16 tends to use up more space than UTF-8 when applied to character sets in which one character always translates to one byte.

Because of how Atlas Hub encodes data, non-English data is human-readable from clients, but not from Microsoft SQL Server and Microsoft SQL Server Express.

Creating a Server Configuration (for an Existing Database)

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create an Atlas Hub configuration using a previously created Microsoft SQL Server or Microsoft SQL Server Express database.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

To create a server configuration using an existing database:

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field. If you want the server configuration to have the same name as the database (a nice convention, especially if you have several server configurations), you must follow the database naming conventions explained at the beginning of this section.
   b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
   c) Select **Microsoft SQL Server/SSE** (the default) from the **Database Type** list.
   d) Uncheck the option to **Create new StarTeam database**, so that Atlas Hub will not automatically create the database for it.
   e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

   f) Click **Next**, and enter the **Database Server name**, **Database name**, **Database login name**, and password in the appropriate text boxes.
   g) Optionally, if you are using a port other than the default, check **Edit Database Port** and type the port number in the text field.

h) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.

i) Click **Finish**. This action re-displays the **Server Administration** tool, which shows your new server configuration as a child of the `Local` node.

Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

4. By default, all server configurations are set to use the TCP/IP endpoint (port) 49201. However, each server configuration on a given computer must have a unique endpoint so it is recommended that you edit the default endpoint. To change the endpoint:
   a) Select the server configuration.
   b) Click the **Start with Override** button (or click **Actions** > **Start with Override** from the main menu). The **Start with Override** dialog box opens.
   c) Enter the endpoint that you want to use in the **TCP/IP Endpoint** field, and click **OK**.
5. Be sure to configure your new server configuration (for information, see the *Server Administration Tool Help*) and plan a backup schedule for it.

Using a Microsoft SQL Server Database

If you will be using a Microsoft SQL Server database, you must complete the following tasks:

1. Install Microsoft SQL Server.

   Important: When you install a Microsoft SQL Server database instance, it defaults to the collation for the locale set for the operating system. This locale setting should be used so long as it is correct for your team. Otherwise, when you automatically create a database from Atlas Hub, you cannot provide a database name, user name, or password in your language.
2. Install Atlas Hub.
3. Create and start an Atlas Hub configuration.
   - If you want to automatically create the database, see *Creating and Starting a Server Configuration*.
   - Otherwise, see *Creating a Database Manually*.

   If you plan to use a supported version of Microsoft SQL Server, see *Connecting to Microsoft SQL Server Databases*.

   Caution: Please note the following:
   - Never modify the database contents other than through the Atlas Hub client or the Server Administration Tool. Direct database manipulation is not supported.
   - Never modify vault files other than through the Atlas Hub client or the Server Administration Tool.

Connecting to Microsoft SQL Server Databases

Atlas Hub requires Microsoft SQL Server authentication to connect to Microsoft SQL Server databases, rather than the default Microsoft Windows authentication. If you install Microsoft SQL Server using the default settings for security and authentication, Atlas Hub will experience problems connecting to the database.

To specify the security to use when connecting to Microsoft SQL Server:

1. Start the Microsoft SQL Server Installation Wizard.
2. Go to the **Authentication Mode** page of the wizard.
3. Select **Mixed Mode** (**Windows Authentication and SQL Server And Windows** option button (instead of the **Windows Only** option).

4. Type and retype the password to use.
5. Click **Next** to complete the rest of the wizard.

Creating and Starting a Server Configuration

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create an Atlas Hub configuration and start it for the first time. It assumes that you want Atlas Hub to automatically create a Microsoft SQL Server or Microsoft SQL Server Express database. If that is not the case, see *Creating a Database Manually*.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

To create a server configuration using an existing database:

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field. If you want the server configuration to have the same name as the database (a nice convention, especially if you have several server configurations), you must follow the database naming conventions explained at the beginning of this section.
   b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
   c) Select **Microsoft SQL Server/SSE** (the default) from the **Database Type** list.
   d) Check the option to **Create new StarTeam database**, so that Atlas Hub automatically creates the database.
   e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

   f) Click **Next**.
4. Enter the server and database information:

5. Enter the information for creating the data files and transaction logs, and click **Finish**.

   Note: We recommend keeping the data files and transaction log files on different physical drives under their own disk controllers.

   The default settings are appropriate for your use if you have fewer than 15 users and expect to store 1GB or less data.

   If you are very familiar with Microsoft SQL Server and Microsoft SQL Server Express databases, you may choose to make some changes by first clearing the **Use Default Configuration** check box and then altering sizes and locations for data files and log files.

   Use at least 3 data files and at least 3 transaction log files when creating a database, because Microsoft SQL Server and Microsoft SQL Server Express databases use a proportional fill strategy. This way all the files tend to become full at about the same time.

   To avoid fragmentation, make the data files as large as possible, based on the maximum amount of data expected in the database.

   The **Server Administration** tool displays your new server configuration as a child of the `Local` node.

   Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

6. By default, all server configurations are set to use the TCP/IP endpoint (port) 49201. However, each server configuration on a given computer must have a unique endpoint so it is recommended that you edit the default endpoint. To change the endpoint:
   a) Select the server configuration.
   b) Click the **Start with Override** button (or click **Actions** > **Start with Override** from the main menu). The **Start with Override** dialog box opens.
   c) Enter the endpoint that you want to use in the **TCP/IP Endpoint** field, and click **OK**.
7. Be sure to configure your new server configuration (for information, see the *Server Administration Tool Help*) and plan a backup schedule for it.

Windows Authentication Support for Microsoft SQL Server

The Atlas Hub database administrator should follow these steps to set up Windows authentication for SQL Server:

1. Create a domain service for Atlas Hub admin. For example: `service.starteamadmin`
2. Login to the Atlas Hub machine as an administrator and add that account to the administrator group.
3. Shutdown the Atlas Hub.
4. Login to Microsoft SQL Server database machine and add the service account to Microsoft SQL Server and change the db ownership to this account using the following script. The following script adds login `service.starteamadmin` to Microsoft SQL Server.

```
Login as sa
Use master
GO
Create LOGIN [<domain name>\service.starteamadmin] FROM WINDOWS WITH
DEFAULT_DATABASE=<starteam database name>
GO
Use <starteam database name>
GO
Exec sp_changedbowner [<domain name>\server.starteamadmin]
GO
```

   Replace `<starteam database name>` with the Atlas Hub database name and `<domain name>` with the domain name.
5. Log back in to Atlas Hub production machine using the Atlas Hub service account.

a) Go to Server Administration and open the configuration properties.
b) Click on the **Database Connection** tab.
c) Check the box **Use Windows Authentication** then click **Verify** to verify database connection.
d) Click **OK**.
e) Click the menu option **Actions** > **Set to run as service**.
f) Uncheck the `localsystem` account and define the service using Atlas Hub service account.

6. Make sure the user account settings are set to `Never Notify`.
7. Start the server.

Guidelines for Data Files and Transaction Logs

Based on the number of users, we suggest the following guidelines for data files and transaction logs. Your needs may be different from those shown in the following tables.

Number of Users	Number of Data Files	Size of Each Data File
Up to 15	3	50 MB
Between 15 and 50	3	300 MB
Between 51 and 100	5	300 MB
Between 101 and 300	7	500 MB
>300	7	800 MB

Number of Users	Number of Log Files	Size of Each Log File*
Up to 15	3	50 MB
Between 15 and 50	3	300 MB
Between 51 and 100	5	300 MB
Between 101 and 300	5	500 MB
>300	6	500 MB

* The transaction log file sizes are relevant only if the transaction log backup is performed frequently.

Transaction log backups are essential. After a transaction is backed up, Microsoft SQL Server and Microsoft SQL Server Express databases automatically truncate the inactive portion of the transaction log. This inactive portion contains completed transactions and is no longer used during the recovery process. The basic advantage comes with the fact that Microsoft SQL Server reuses this truncated, inactive space in the transaction log instead of allowing the transaction log to continue to grow and use more space. This is a huge plus from a performance standpoint.

Allowing files to grow automatically can cause fragmentation of those files if a large number of files share the same disk. Therefore, it is recommended that files or file groups be created on as many different available local physical disks as possible. Place objects that compete heavily for space in different file groups.

Running SQL Scripts for Microsoft Databases

Atlas Hub comes with some SQL scripts written specifically for use with your database. These scripts help you maintain and tune Atlas Hub databases. You run some SQL scripts after installation, some on a weekly basis for database performance maintenance, and some scripts are run for you automatically by Atlas Hub.

The SQL scripts for Microsoft SQL Server and Microsoft SQL Server Express databases that you may run are located in the `Micro Focus\Hub\DBScripts\Sqlserver_Scripts` folder.

Note: The `Sqlserver_Scripts` folder contains several subfolders: `Create_Stored_Procedures`, `Drop_Stored_Procedures`, `Install`, `Preinstall`, `Generic`,

and `DW` (for Data Warehouse). The scripts in these subfolders are run by Atlas Hub as needed. Never execute any of them directly from an external database user interface, such as SQL Query Editor.

The following table lists the SQL scripts that you are most likely to need. Some should be run manually on a regular basis. The table recommends the frequency for running these scripts. You may adjust the frequency depending on the Atlas Hub usage at your facility. Run scripts at times when the server is least used, such as overnight or on weekends.

In addition to running these scripts, you should also run a **Purge** option from the **Server Administration** tool to remove deleted views from the database. We recommend purging the database after you have deleted one or more views from a project. See the *Server Administration Tool Help* for information on the **Purge** option.

Atlas Hub Script Name	Run Frequency
starteam_sqlserver_dbcc.sql	Weekly
starteam_sqlserver_dbcc_reindex.sql	Twice a week (minimum)
starteam_sqlserver_dbcc_showcontig.sql	Twice a week (minimum)
starteam_sqlserver_dropall.sql	Only if necessary

Caution:

- Before running any of the Atlas Hub SQL scripts for a Microsoft SQL Server or Microsoft SQL Server Express database, ensure that the database compatibility mode is set correctly. For Microsoft SQL Server 2008-based configurations, set the database compatibility mode to 90.
- Be sure to backup your Atlas Hub database, as necessary, and verify these backups periodically. You should restore and test backups of your Atlas Hub project data on a test system. Restoring and testing your backups helps to ensure that your data is being backed up correctly.

To run a script for a Microsoft SQL Server or Microsoft SQL Server Express database:

1. Install **SQL Server Management Studio** or **SQL Server Management Studio Express** from Microsoft. Enter the result of your step here (optional).
2. Click **Start** > **Microsoft SQL Server [or Microsoft SQL Server Express]** > **SQL Server Manager Studio [or SQL Server Manager Studio Express]**.
3. Design a new query or open an existing one in **SQL Server Manager Studio**.
4. Click **Query** > **Connection** > **Connect** to connect to the server that contains the database you want to access.
5. Select the appropriate Atlas Hub database.
6. Open the tuning script, by choosing **File** > **Open** > **foldername\scriptname**.
7. Execute the script, by clicking the **Execute** button on the toolbar or by pressing **F5**.

SQL Scripts for Microsoft SQL Server and SSE Databases

`starteam_sqlserver_dbcc.sql`

Run: weekly.

This script rebuilds the database indexes and performs a consistency check on the database objects. This script builds the indexes and updates the statistics in the database distribution pages.

`starteam_sqlserver_dbcc_reindex.sql`

Run: at least twice a week.

This script rebuilds all the indexes in the database. It is extremely important to run this script routinely.

`starteam_sqlserver_dbcc_showcontig.sql`

Run: at least twice a week.

This script gives information on database fragmentation.

`starteam_sqlserver_dropall.sql`

Run: only if necessary.

Caution: Running this script deletes all Atlas Hub tables and the data they contain from the database. Use this script with extreme caution.

One use case example is if you migrate an Atlas Hub configuration to another database, you might use this script to remove tables from the original database. Another example is if you mistakenly add the Atlas Hub tables to a tablespace other than the Atlas Hub tablespace, use this script to remove them.

Microsoft SQL Server Security

SQL Server uses a standard security model that involves the following entities and concepts:

Securable	Represents a resource or target object that requires securing, such as a database view.
Principal	Represents a user who requests access to a resource.
Permission	Access type that is associated with securable. Permissions can be granted to or revoked from principals. For example, `Update` is a permission that is associated with a table (`securable`) named `R`. If `Update` on `R` is granted to a user (`principal`) named `U`, then `U` receives `Update` access on `R`.

Further, Microsoft SQL Server supports the following security principals at different levels:

Windows-level principals	Control access to SQL Server instances for Windows Local Login and Windows Network Domain Login.
SQL Server-level principals	Control access to Microsoft SQL Server instances for SQL Server Login.
Database-level principals	Control access to database instances for database users.

To access a Microsoft SQL Server instance, use a Microsoft Windows user name or a Microsoft SQL Server user name that was previously created in that server instance. After you log on, the user name represents you as your security principal at the server level.

If you try to use a specific database in the server, Microsoft SQL Server searches the appropriate database for any previous user who has been mapped to your user name. If Microsoft SQL Server locates such a user, the corresponding user name represents you as your security principal at the server level.

Creating a Database Manually

Despite the fact that Atlas Hub has automated Microsoft SQL Server and Microsoft SQL Server Express database creation, you may prefer to create your own. This makes more sense for Microsoft SQL Server because there are good tools for database creation.

It is very important that you use the directions in the following procedure.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

To create your own Microsoft SQL Server or Microsoft SQL Server Express database:

1. Install Microsoft SQL Server or Microsoft SQL Server Express. If you plan to use a supported version of Microsoft SQL Server, be sure to review the section *Connecting to Microsoft SQL Server Databases*.
2. Install Atlas Hub.
3. Create an Atlas Hub database. Contact your database administrator about the specifics. Be sure that:
   - The database is owned by an Atlas Hub user.
   - The default database for the Atlas Hub user is the Atlas Hub database.
   - The database will fit the expected growth patterns for storing your Atlas Hub data. See *Guidelines for Data Files and Transaction Logs*.
   - The name of the database follows the conventions explained earlier in this section.
   - To create or upgrade the Atlas Hub database, the Microsoft SQL Server instance must have one of the following collations:
     - `Latin1_General_CI_AI`
     - `SQL_Latin1_General_CP1_CI_AS`

     On a Japanese double-byte operating system, where the default collation for the Microsoft SQL Server installation is not supported by Atlas Hub , you must use a named instance with one of the supported collations.

     Note: Microsoft SQL Server's multi-instance feature supports the use of multiple instances in different locales on the same database server.
4. Create and start an Atlas Hub configuration. See *Creating a Server Configuration (for an Existing Database)*.

Understanding Collation

The physical storage of character strings in the supported versions of Microsoft SQL Server and Microsoft SQL Server Express databases is controlled by collations. A collation specifies the bit patterns that represent each character and the rules by which characters are sorted and compared.

Microsoft SQL Server supports objects that have different collations being stored in a single database. Separate Microsoft SQL Server collations can be specified down to the level of columns. Each column in a table can be assigned different collations.

In a computer, characters are represented by different patterns of bits being either `ON` or `OFF`. A program that uses one byte (eight bits) to store each character can represent up to 256 different characters. A program that uses two bytes (16 bits) can represent up to 65,536 characters.

Single-byte code pages are definitions of the characters mapped to each of the 256 bit patterns possible in a byte. Code pages define bit patterns for uppercase and lowercase characters, digits, symbols, and special characters such as `!, @, #, or %`. Each European language, such as German or Spanish, has its own single-byte code page. Although the bit patterns used to represent the Latin alphabet characters A through Z are the same for all the code pages, the bit patterns used to represent accented characters (such as é and á) vary from one code page to the next. If data is exchanged between computers running different code pages, all character data must be converted from the code page of the sending computer to the code page of the receiving computer. If the source data has extended characters that are not defined in the code page of the receiving computer, data is lost. When a database serves clients from many different countries, it is difficult to pick a code page for the database that contains all the extended characters required by all the client computers. Also, a lot of processing time is spent doing the constant conversions from one code page to another.

Single-byte character sets are also inadequate to store all the characters used by many languages. For example, some Asian languages have thousands of characters, so they must use two bytes per character. Double-byte character sets have been defined for these languages. Still, each of these languages have

their own code page, and there are difficulties in transferring data from a computer running one double-byte code page to a computer running another.

For information about synchronizing collation settings with another Windows locale, see the following Microsoft site: *http://msdn2.microsoft.com/en-us/library/aa176553.aspx*.

How Is the Default Collation Selected?

Microsoft SQL Server Setup chooses the Windows collation that supports the Windows locale of the computer on which the instance of Microsoft SQL Server is being installed. If the computer is using the US English locale, the instance's default collation is `Latin1_General_CI_AS`.

Important: On a Japanese double-byte operating system, where the default collation for the Microsoft SQL Server installation is not supported by Atlas Hub, you must use a named instance with one of the following supported collations.

- `Latin1_General_CI_AI`
- `SQL_Latin1_General_CP1_CI_AS`

The multi-instance feature of supported Microsoft SQL Server versions supports the use of multiple instances in different locales on the same database server.

Backing up Microsoft SQL Server Databases

This section outlines the backup options available to DBAs and makes recommendations for backing up the databases used by the server configurations. Be aware that these are just recommendations. Any finalized disaster recovery plan must be created by your organization in consultation with its IT infrastructure staff.

An application backup consists of backing up both the database and the application archive files and you have a choice between online and offline backups. If all of your archive files are in Native-II format, you can back up a server configuration online, without shutting it down or locking it.

For server configuration online backups, it is essential to take full database and transaction log backups. The entire database can be recreated from a database backup in one step by restoring the database. The restore process overwrites the existing database or creates the database if it does not exist. The restored database will match the state of the database at the time the backup completed, minus any uncommitted transactions. Uncommitted transactions are rolled back when the database is recovered.

Based on the resource requirements, the DBA can also choose the recovery model for the database. The recovery model balances logging overhead against the criticality of fully recovering the data. The recovery models supported by Microsoft SQL Server are:

Full	The data is critical and must be recoverable to the point of failure. All data modifications are logged. All Microsoft SQL Server recovery options are available.
Bulk-logged	Certain bulk operations, such as bulk copy operations, `SELECT INTO`, and text processing, can be replayed if necessary, so these operations are not fully logged. You can recover only to the end the last database or log backup.
Simple	All data modifications made since the last backup are not available. This type of recovery scenario has the lowest logging overhead, but cannot recover past the end of the last backup.

Microsoft SQL Server supports the following types of backups:

Full database backup

A full database backup creates a duplicate of the data that is in the database. This is a single operation, usually scheduled at regular intervals. Full database backups are self-contained. Full backups provide a snapshot of the database. Most of the recovery options require a full backup to be present.

We strongly recommend the use of full backups.

Differential backup

A differential database backup records only the data that has changed since the last database backup. Frequent differential backups are recommended to reduce backup times. Making frequent backups decreases the risk of losing data.

Differential backups restore the data that they contain to the database. Differential backups cannot be used to recover the database to a point in time.

The availability of a differential backup minimizes the time it takes to roll forward transaction log backups when restoring a database.

Transaction log backup

The transaction log is a serial record of all the transactions that have been performed against the database since the transaction log was last backed up. With transaction log backups, you can recover the database to a specific point in time or to the point of failure.

When restoring a transaction log backup, Microsoft SQL Server rolls forward all the changes recorded in the transaction log. When Microsoft SQL Server reaches the end of the transaction log, it has re-created the exact state of the database at the time of the backup operation.

If the database is recovered, Microsoft SQL Server then rolls back all transactions that were incomplete when the backup operation started. Transaction log backups generally use fewer resources than database backups. As a result, you can create them more frequently than database backups. Frequent backups decrease the risk of losing data. For high volume Online Transaction Processing (OLTP) environments, it is desirable to create transaction log backups more frequently.

Transaction log backups can only be used with Full and bulk-logged recovery models. The transaction log cannot be backed up during a full database backup or a differential database backup. However, the transaction log can be backed up while a file backup is running.

Never backup a transaction log before a database backup is created because the transaction log contains the changes made to the database after the last backup was created.

Never truncate the transaction log manually because it breaks the backup chain. If a transaction log has been truncated, take a full database backup to start a new backup chain.

File backups

A file or file group backup consists of the backing up of individual data files (or the files in the file group). A file-based recovery model increases the speed of recovery by allowing you to restore only the damaged files without restoring the rest of the database. For example, suppose a database is comprised of several files located physically on different disks and one disk fails. Only the file on the failed disk needs to be restored and rebuilt using the transaction log backup.

File backup and restore operations must be used in conjunction with transaction log backups. For this reason, file backups can only be used with the full recovery and bulk-logged recovery models.

Recommendations

We recommend that you:

- Use the full recovery model.
- Perform a full database backup once every day. For full database sizes greater than 3 GB, it is okay to perform full backups on alternate days. If you perform full backups on alternate days, we strongly recommend that you create daily differential backups.
- Create daily transaction log backups after the completion of the full or differential backup. In addition to this, schedule a transaction log backup every 4 hours. Never truncate a transaction log manually.
- In case of a disaster, create a backup of the currently active transaction log. If active transaction log backup is not available (for example, because a media failure for the drive hosting the transaction logs and drive is not being mirrored), the database cannot be recovered past the last available transaction

log backup. This would hamper a point-in-time recovery beyond the last available transaction log backup.

- Label the backup media correctly.
- Keep backup copies in offsite locations.

Configuring Oracle Databases

This chapter explains how to create a new server configuration and Oracle schema user for use with Atlas Hub. Atlas Hub stores everything except for file archives and server startup information in the schema user.

This chapter also provides an overview of the tuning and maintenance SQL scripts that are provided with Atlas Hub and explains how to use them.

When you create a server configuration, Atlas Hub can automatically create the tablespace and schema user. However, if you prefer, you can create the tablespace and schema user manually.

Note: All of the tablespaces created using Atlas Hub automatically create locally-managed tablespaces. All tablespaces created by Atlas Hub use AUTOALLOCATE. Atlas Hub supports Oracle Universal Installer for client. Consult with your database administrator for more information.

Terminology

When this chapter uses the terms "database instance" and "schema user", it uses Oracle terminology. Each database instance can manage a number of different schema users. Each Atlas Hub configuration has its own schema user. Elsewhere in the documentation you will find the terminology common to the other databases that Atlas Hub supports. For example, when you install Microsoft SQL Server on a computer, you can run several instances of it. Each instance can manage a number of different databases. Therefore, when you see the term “database” in other parts of these manuals, think “schema user”.

Logging Onto Schema Users

It is highly recommended that you use a dedicated Atlas Hub user account to run or log onto Atlas Hub schema users. System administrator accounts usually have unlimited privileges. Any anomalies or errors that occur while logged in as the system administrator may result in unrecoverable damage to the Atlas Hub schema users and other schema users managed by the same database.

Using an Oracle Schema User

If you will be using an Oracle schema user, you must complete the following tasks:

1. Install Oracle Server. Most customers install Atlas Hub and Oracle Server on separate computers.
2. Install Oracle Client on the computer on which you plan to install Atlas Hub.
3. Establish connectivity between Oracle Client and Oracle Server. See *Verifying Connectivity Between Client and Server*.
4. Install Oracle Enterprise Manager.
5. Install Micro Focus Atlas.
6. Make sure that the NLS_LANG setting for Oracle Client and the character set specified for Oracle Server are correct. For more information, see *Using NLS_LANG Correctly* and *Database Character Set*.
7. Create an Atlas Hub configuration. For more information, see *Creating and Starting a Server Configuration with an Oracle Schema User* (about automatically created schema users) and *Creating the Oracle Schema User Manually*.

   Caution: Please note the following:

   - Never modify the database contents other than through the Atlas Hub client or the Server Administration Tool. Direct database manipulation is not supported.
   - Never modify vault files other than through the Atlas Hub client or the Server Administration Tool.

Verifying Connectivity Between Client and Server

If Oracle Client and Oracle Server are located on different computers (as recommended), verify that Oracle Client can connect with the Oracle database instance. This ensures that an Atlas Hub configuration can access the Oracle schema user.

To verify the connection between Oracle Client and Oracle Server:

1. Using Oracle Client on the Atlas Hub computer, set up Oracle connectivity.
2. Use **Net Configuration Assistant** to establish the connectivity between the computers on which Oracle Client and Server are installed.
3. Test the database connectivity through **SQL*Plus** or **SQL*Worksheet**.

Creating a Server Configuration for an Oracle Schema User

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create an Atlas Hub configuration and start it for the first time.

Before you begin, you need to know:

- The Oracle Database Server name and either the Oracle Service name or SID.
- The system password.
- The logon name and password for the schema user.

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field.
   b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
   c) Select `Oracle` from the **Database Type** list.
   d) Uncheck the option to **Create new StarTeam database**, so that Atlas Hub will not automatically create the database for it.
   e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

   f) Click **Next**, and enter the **Database Server name**, **Database name**, **Database login name**, and password in the appropriate text boxes.
   g) Optionally, if you are using a port other than the default, check **Edit Database Port** and type the port number in the text field.

h) Type the login name and password for the schema user in the appropriate field.
i) Select either **Service Name** or **SID** and provide the information in the appropriate field.
j) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.
k) Click **Finish**.

This action re-displays the **Server Administration** tool, which shows your new server configuration as a child of the `Local` node.

Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

Installing Oracle Client

Oracle Client is required to use Oracle with Atlas Hub. A list of choices for various operating systems is available at: *http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html*.

1. Select **See All for Windows 32 bit or Windows x64** to display downloads specific to that OS.
2. Install the driver that matches the version of Atlas Hub that you plan to use (32 or 64 bit).
3. Choose **Oracle Database 11g Release 2 Client**.
4. During the Oracle Client installation, there are various options. Atlas Hub only requires the **Instant Client**. Most Oracle installation packages (Universal Installer, SQL Developer) install the appropriate Oracle drivers on a given system. If you have already installed these products, it is likely that no other Oracle driver related installation is necessary.

Using NLS_LANG Correctly

Atlas Hub converts data that is stored in the database to UTF-8 format. Your database character set should be defined such that it will not conflict with this nor with other Atlas Hub requirements. For more information, see *Defining NLS_LANG in the Windows Registry*.

`NLS_LANG` represents the Oracle Client character set. The Oracle Client character set is defined by the third part of the `NLS_LANG` parameter and must be set to a value that Server uses.

After setting `NLS_LANG` correctly, Oracle Client correctly translates any symbols to the character code values of the database character set when storing data into the database or retrieving data from the database. `NLS_LANG` is specified in the following format.

```
NLS_LANG = language_territory.characterset
```

language	Specifies the language used for Oracle messages, day names, and month names.
territory	Specifies the monetary and numeric formats, territory, and conventions for calculating week and day numbers.
characterset	Controls the character set used by Oracle Client. (Normally it matches your Microsoft Windows code page.)

NLS_LANG Definition Overview

The ODBC installation supports multiple Oracle homes. Each ODBC driver is uniquely identified by the name of the Oracle home in which it is installed. Applications using Oracle ODBC drivers use the value of NLS_LANG for making decisions related to character set conversion. The character set portion of the `NLS_LANG` setting must be `AMERICAN_AMERICA.WE8MSWIN1252`. If your database was previously Oracle 8i, then you use `AMERICAN_AMERICA.WE8ISO8859P1`.

The ODBC driver installation uses the value of the Windows code page (ACP) and defines the value of `NLS_LANG` in the registry. The value of `NLS_LANG` should be verified for accuracy. Be sure to check for the

NLS_LANG environment variable because it will override the registry setting. You may choose to delete the variable. For more information, see *Defining the NLS_LANG as an Environment Variable*.

You can define NLS_LANG for the Oracle Client in either of the following ways:

In the Registry (Microsoft Windows only)	Oracle Client and Server support the concept of multiple Oracle homes. What this means is that you can have multiple sets of Oracle binaries on the same computer. When you have multiple Oracle homes on the same computer, you have multiple registry entries for each home. NLS_LANG should be defined for each home.
As an environment variable	Although the Registry is the primary repository for environment settings, it is not the only place where environment parameters can be set. You can set the NLS_LANG as a System or User Environment Variable in the System properties. **Caution:** If NLS_LANG is defined as an environment variable, it will be used for all Oracle homes. This option should be used with caution, especially for cases where there are multiple Oracle homes on the computer and some Oracle homes have different NLS_LANG settings. It may be best to delete it.

Defining NLS_LANG in the Microsoft Windows Registry

Note: If NLS_LANG is not defined, it defaults to AMERICAN_AMERICA.US7ASCII and must be changed to a value that Server uses.

To change the NLS_LANG setting using the Microsoft Windows **Registry Editor**:

1. Click **Start** > **Run**. The **Run** dialog box opens.
2. Type `regedit`, and then click **OK**.
3. Edit the following registry entry: HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\homeID where homeID is the unique name identifying the Oracle home.

Defining the NLS_LANG as an Environment Variable

Note: If NLS_LANG is not defined, it defaults to AMERICAN_AMERICA.US7ASCII and must be changed to a value that Server uses.

Note: To create or modify this setting for LINUX, define the value of NLS_LANG in the `.profile` or `.cshrc` file. Contact your LINUX Admin to learn how to set environment variables.

To create or modify the NLS_LANG environment variable's setting on Microsoft Windows:

1. Right-click the **My Computer** icon on your desktop, and then choose **Properties** from the context menu.
2. Select the **Advanced** tab.
3. Click **Environment Variables**. The **Environment Variables** dialog box opens.
4. Select NLS_LANG and define an appropriate value.

Database Character Set and Atlas Hub

Atlas Hub stores UTF-8 data in Oracle schema users. Atlas Hub does not use the Oracle conversion mechanism. It performs the conversion to UTF-8 itself.

Caution: Because Atlas Hub does the conversion, it is very important to prevent Oracle Client from also converting the data. A double conversion will result in garbage.

Oracle software does the conversion only when the Oracle Client's character set (specified by NLS_LANG) does not match the database instance's character set. Therefore, it is very important that the client and server's settings match. Atlas Hub configurations require that Oracle database instances use the WE8MSWIN1252 character set. If your Atlas Hub database was previously Oracle 8i, then Oracle database instances use the WE8ISO8859P1 character set.

The **Server Administration** tool does not allow you to create a new Atlas Hub server configuration unless the underlying database instance's character set is the correct character set. When Atlas Hub starts, it also ensures that the database character set matches the characters set portion of `NLS_LANG`.

Due to the numeric setting differences between different languages and territories, Atlas Hub also does not start unless `AMERICAN_AMERICA` (the US numeric setting) is the setting for the `language_territory` portion of `NLS_LANG`. Failure to do so causes exceptions for Atlas Hub. This does not mean that numbers and so on will be displayed incorrectly. For example, in Germany and other countries, a decimal point is represented as a comma. Such numbers will still display correctly.

Creating and Starting a Server Configuration with an Oracle Schema User

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create an Atlas Hub configuration and start it for the first time. It assumes that you want Atlas Hub to automatically create the Oracle tablespace and a schema user named StarTeam. If that is not the case, see *Overview of Creating the Oracle Schema User Manually*.

For new configurations, you must know the Oracle Database Server name and either the Oracle Service name or SID.

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field.
   b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
   c) Select `Oracle` from the **Database Type** list.
   d) Check the option to **Create new StarTeam database**, so that Atlas Hub automatically creates the tablespace and schema user for the schema user.
   e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

   f) Click **Next**. The **Create an Oracle Schema User** page of the wizard opens.
   g) Type the Oracle Server name or SID in the appropriate field.
   h) Optionally, if you are using a port other than the default, check **Edit Database Port** and type the port number in the text field.
   i) Enter the database system password in the **System password** field.
   j) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.

k) To keep the name of the server configuration and the schema user the same, both the **New schema user name** field default to the name you provided earlier for the server configuration. Change these names if you prefer to use different values.

l) Enter and confirm a password for the schema user name.

m) Click **Next**. The **Create a Tablespace for StarTeam Schema** page of the wizard opens.

n) The tablespace name defaults to the name of your server configuration. Change this name if you prefer to use a different value.

o) If you have fewer than 15 users and expect to store 1 GB or less of data, the default settings are appropriate for your use. If you are very familiar with Oracle schema users, you may choose to alter the names, sizes, and locations of the data files. For more information, see *Guidelines for Data Files*. To avoid fragmentation, make the data files as large as possible, based on the maximum amount of data expected in the database. Use at least three data files when creating a tablespace because:

- There is a size limit of 2 GB per data file.
- Fewer files can result in slow response times when insert activity is heavy.

p) Click **Finish**.

This action re-displays the **Server Administration** tool, which shows your new server configuration as a child of the `Local` node.

Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

4. By default, all server configurations are set to use the TCP/IP endpoint (port) 49201. However, each server configuration on a given computer must have a unique endpoint so it is recommended that you edit the default endpoint. To change the endpoint:

a) Select the server configuration.

b) Click the **Start with Override** button (or click **Actions** > **Start with Override** from the main menu). The **Start with Override** dialog box opens.

c) Enter the endpoint that you want to use in the **TCP/IP Endpoint** field, and click **OK**.

5. Be sure to configure your new server configuration (for information, see the *Server Administration Tool Help*) and plan a backup schedule for it.

Guidelines for Data Files

We suggest the following guidelines for the number of data files and their sizes, based on the number of users. Your needs may be different than those shown in the table below.

Number of Users	Number of Data Files	Size of Each Data File
Up to 15	3	50 MB
Between 15 and 50	3	300 MB
Between 51 and 100	5	300 MB
Between 101 and 300	7	500 MB
>300	7	800 MB

Running SQL Scripts for Oracle Schema Users

Atlas Hub comes with some SQL scripts written specifically for use with your database. These scripts help you maintain and tune Atlas Hub databases. You run some SQL scripts after installation, some on a weekly basis for database performance maintenance, and some scripts are run for you automatically by Atlas Hub.

The SQL scripts for Oracle schema users that you may run are located in the `Micro Focus\Hub \DBScripts\Oracle_Scripts` folder.

Note: The `Oracle_Scripts` folder contains several subfolders: `Create_Stored_Procedures`, `Drop_Stored_Procedures`, `Install`, `Preinstall`, `Generic`, and `DW` (for Data Warehouse). The scripts in these subfolders are run by Atlas Hub as needed. Never execute any of them directly from an external database user interface, such as **SQL*Plus** or **SQL *Worksheet**.

The following table lists the SQL scripts that you are most likely to need. Some should be run manually on a regular basis. The table recommends the frequency for running these scripts. You may adjust the frequency depending on the Atlas Hub usage at your facility. Run scripts at times when the server is least used, such as overnight or on weekends.

In addition to running these scripts, you should also run a **Purge** option from the **Server Administration** tool to remove deleted views from the database. We recommend purging the database after you have deleted one or more views from a project. See the *Server Administration Tool Help* for information on the **Purge** option.

StarTeam Script Name	Run Frequency
`starteam_oracle_compute_stats.sql`	Weekly
`starteam_oracle_dropall.sql`	Only if necessary
`starteam_oracle_rebuild_indexes.sql`	Weekly

Caution: Be sure to backup your Atlas Hub schema user, as necessary, and verify these backups periodically. You should restore and test backups of your Atlas Hub project data on a test system. Restoring and testing your backups helps to ensure that your data is being backed up correctly.

To run a SQL script for Oracle schema users:

1. Go to the command prompt.
2. Change directories to the directory containing the Atlas Hub SQL scripts for Oracle schema users.
3. At the command prompt, type: `sqlplus username/password@servicename`

   username The Atlas Hub Oracle Schema User Name.

   password The Atlas Hub Oracle Schema Password.

   servicename The Net Service Name created using Oracle Net 8 Easy Config.
4. Execute the script.

   For example, to execute the `starteam_oracle_compute_stats.sql` script, type `@starteam_oracle_compute_stats.sql` and press `Enter`.

Atlas Hub SQL Scripts for Oracle Schema Users

starteam_oracle_compute_stats.sql

Run: weekly.

The `starteam_oracle_compute_stats.sql` script updates the statistics in the database distribution page for all tables in the database. This data enables the query optimizer to choose the right index for a given query.

starteam_oracle_dropall.sql

Run: only if necessary.

Caution: Running this script deletes all Atlas Hub tables and the data they contain from the database. This script can be executed from either Sql*Plus or SQL*Worksheet. Use this script with extreme caution.

One use case example is if you migrate an Atlas Hub configuration to another database, you might use

	`starteam_oracle_dropall.sql` to remove tables from the original database. Another example is if you mistakenly add the Atlas Hub tables to a tablespace other than the Atlas Hub tablespace, use the `starteam_oracle_dropall.sql` script to remove them.
starteam_oracle_rebuild_indexes.sql	Run: weekly. The `starteam_oracle_rebuild_indexes.sql` script rebuilds the database indexes and configures the storage parameters for the index tablespace. The script assumes that the indexes are located in a tablespace named `INDX`. If your index tablespace uses a different name, `edit starteam_oracle_rebuild_indexes.sql` to reflect the correct tablespace name. Run the `starteam_oracle_rebuild_indexes.sql` script weekly to enhance database data retrieval.

Creating the Oracle Schema User Manually

This section explains how to manually create an Oracle tablespace and schema user for Atlas Hub. It also provides an overview of the tuning and maintenance SQL scripts that are provided with Atlas Hub, and explains how to use them. For detailed information on Oracle performance tuning, consult your Oracle documentation.

To configure an Oracle database instance:

1. Complete the procedures described in *Using an Oracle Schema User*.
2. Create an Atlas Hub tablespace.
3. Create an Atlas Hub schema user.
4. To ensure that Atlas Hub can create labels and perform other operations successfully, make sure that the schema user has the following Oracle privileges. They must be explicitly defined rather than relegated to roles. For more information, see *Granting Oracle Privileges*.
   - `Create procedure`
   - `Create sequence`
   - `Create session`
   - `Create table`
   - `Create trigger`
   - `Create type`
   - `Create view`
   - `Unlimited tablespace`
   - `Create any context`
   - `Drop any context`
5. Create the server configuration. For more information, see *Creating a Server Configuration for an Oracle Schema User*.

Backing Up Oracle Databases

This section outlines the backup options available to DBAs and makes recommendations for backing up the databases used by the server configurations. Be aware that these are just recommendations. Any finalized disaster recovery plan must be created by your organization in consultation with its IT infrastructure staff.

An application backup consists of backing up both the database and the application archive files and you have a choice between online and offline backups. If all of your archive files are in Native-II format, you can back up a server configuration online, without shutting it down or locking it.

An online or hot backup is a backup performed while the database is online and available for read/write operations. Except for Oracle exports, you can only perform online backups when running in ARCHIVELOG mode. An offline or cold backup is a backup performed while the database is offline and unavailable to its users.

Typically an Oracle DBA uses one or more of the following options to back up an Oracle database.

Logical Backups (Export/Import)

Oracle exports are “logical” database backups (not physical) as they extract data and logical definitions from the database into a file. Other backup strategies normally back up the physical data files. One of the advantages of exports is that you can selectively re-import tables. However, you cannot roll forward from a restored export file. To completely restore a database from an export file, you almost need to recreate the entire database. Logical backups takes a snapshot of the database schema as it was at a particular time.

Offline/Cold Backups

A backup performed when the database is shut down is known as an offline or cold backup. You must copy the data files, control file and online redo log files using an OS copy utility. This is a considered a complete backup of the database. Any changes made after this backup are unrecoverable if the database is running in NOARCHIVELOG mode. All transactions are recorded in online redo log files whether the database is archiving or not. When redo logs are archived (ARCHIVELOG mode), Oracle allows you to apply these transactions after restoring files that were damaged (assuming that an active redo log file was not among the files damaged).

Whenever the schema of the database is changed, such as when you add a new data file, rename a file, or create or drop a tablespace, you must shut down the database and copy at least the control file and the newly added data file. A complete backup of the database is preferred.

Before performing a cold backup, it is essential to get a list of all the Oracle files that need to be backed up. Running the following queries will provide a list of all the files.

```
select name from sys.v_$datafile;
select member from sys.v_$logfile;
select name from sys.v_$controlfile;
```

Shut down the database from SQL*Plus or Server Manager. Back up all the files to secondary storage (for example, tapes). Ensure that you back up all data files, all control files, and all log files. When completed, restart your database.

Note: If your database is in ARCHIVELOG mode, you can still use archived log files to roll forward from an offline backup. If you cannot take your database down for an offline backup at a convenient time, switch your database into ARCHIVELOG mode and perform an online backups.

Online/Hot Backups

A backup performed when the database instance is running is known as online or hot backup. Online backups are very important at customer sites where a database instance must operate 24-hours per day and offline backups are not feasible. During the duration of an online backup, the database remains available for both reading and updating. For this kind of backup, the database must be in ARCHIVELOG mode. Only data files and current control files need to be backed up. Unlike offline backups, the unit of a online backup is a tablespace, and any or all tablespaces can be backed up whenever needed. Different data files can be backed up at different times.

To perform an online backup, you switch the tablespace into “backup mode” before copying the files as shown in the following example:

```
ALTER TABLESPACE xyz BEGIN BACKUP;
! cp xyfFile1 /backupDir/
ALTER TABLESPACE xyz END BACKUP;
```

It is better to backup individual tablespaces than to put all tablespaces in backup mode at the same time. Backing them up separately incurs less overhead. After completing the tablespace backups, it is important to back up the control files as shown in the following example.

```
ALTER SYSTEM SWITCH LOGFILE; --Force log switch to update control file headers
ALTER DATABASE BACKUP CONTROLFILE TO '/directory_name/control.dbf';
```

The frequency of online backups is inversely proportional to the time taken for recovery in case of a media failure. The older your backup, the more redo log files need to be applied, and the recovery times increases. Backup strategies should be tested before being used to protect a production database.

We strongly recommend that you run online backups at times when the database is least accessed, during non-peak hours. Oracle writes complete database blocks instead of the normal deltas to redo log files while in backup mode. This leads to excessive database archiving and could lock up the database.

RMAN Backups

Recovery Manager (RMAN) is an Oracle tool that lets the DBA back up and recover Oracle databases. RMAN lets you perform full backups (with the database online or offline), incremental backups on the block level, and backups of online redo logs and control files. The SYSDBA privilege is required to run RMAN on a database. The other benefits of RMAN backups are that you can:

- Keep track of all backup and recovery operations performed against the database.
- Manage centralized backup and recovery procedures for the enterprise.
- Identify corrupt blocks.
- Back up only those blocks that actually contain data. This can lead to significant savings in backup space requirements.
- Have support for incremental backups. Incremental backups back up only those blocks that have changed since a previous backup. This helps with the disk space usage and reduces the backup times significantly. The Oracle 10g feature called “block change training” provides significant improvement for incremental backups. Contact your DBA about how to implement this feature.

The following examples of RMAN backup and restore are extremely simplistic and are included on to illustrate basic concepts. By default, Oracle uses the database control files to store information about backups. Normally, you prefer to set up an RMAN catalog database to store RMAN metadata. Read the Oracle Backup and Recovery Guide before implementing any RMAN backups.

```
rman target sys/*** nocatalog
 run {
  allocate channel t1 type disk;
  backup
   format '/app/oracle/db_backup/%d_t%t_s%s_p%p'
    ( database );
   release channel t1;
 }
Example RMAN restore:
  rman target sys/*** nocatalog
  run {
   allocate channel t1 type disk;
   restore tablespace users;
   recover tablespace users;
   release channel t1;
}
```

Export/Import Data Pump

Oracle introduced the export/import data pump in the 10g release. The import pump is twenty times faster than the conventional import utility. Export/Import data pump utilities are “logical” database backups (not physical) as they extract data and logical definitions from the database into a file. Export/Import data pump utilities do not fit into 24/7 model because they do not offer roll-forward capabilities. Export data pump provides a snapshot of the database schema as it was at a particular time.

Recommendations

We strongly recommend the use of RMAN backups if your enterprise wants to run an Atlas Hub instance in a 24/7 environment. RMAN has evolved over the last few years and Oracle continues to add features that make disaster recovery easier, more reliable, and faster.

Oracle Database Tuning

This section provides the basic information needed to create an Oracle schema for Atlas Hub. We recommend using the **Server Administration** tool to create the schema, but if you prefer to create your own, follow the guidelines provided in this section.

The most efficient way to tune your Oracle database is to start with the recommended database settings and monitor the instance using the advisories. In addition to that, we recommend the use of Automatic Workload Repository (AWR) to collect performance statistics, including wait events used to identify performance problems. A detailed description of AWR is beyond the scope of this document. Please refer to your Oracle 10g performance tuning guide for more information.

Recommended Initialization Parameters

The following two tables recommend Oracle parameter settings for use with Atlas Hub databases.

Table 1: Common Database Configuration Parameters

Parameter	Recommended Value
`Compatible`	10gR2: 10.2.0 11g: 11.1.0.0.0
`Cursor_sharing`	10gR2: Force 11g: Force
`Log_checkpoint_interval`	Greater than the redo log size
`Log_checkpoint_timeout`	0
`Workarea_size_policy`	Auto
`Db_block_size`	16384 (16k)
`Db_file_multi_block_read_count`	16
`Optimizer_mode`	first_rows
`Timed_statistics`	True
`Open_cursors`	400
`Undo_management`	Auto
`Undo_tablespace`	(Name of the undo tablespace)
`Undo_retention`	28800
`Processes`	250
`Statistics_level`	Typical

Table 2: Database parameters based on total memory

Total Memory	Recommended 10gR2 Settings	Recommended 11g Settings
1 GB	SGA_TARGET = (Total Physical Memory * 80%) * 60%.	MEMORY_TARGET = Total Physical Memory * 75%.

Total Memory	Recommended 10gR2 Settings	Recommended 11g Settings
	We assume that 20% of the total memory will be used by the OS. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 524288. PGA_AGGREGATE_TARGET = (Total Physical Memory * 80%) * 30%. 30% of the non-OS available memory. This is the starting value. This may need to be adjusted upwards.	We assume that 20% of the total memory will be used by the OS. The Oracle instance should be running on a dedicated machine. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 524288.
2 GB	SGA_TARGET = (Total Physical Memory * 80%) * 60%. We assume that 20% of the total memory will be used by the OS. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 1048576. PGA_AGGREGATE_TARGET = (Total Physical Memory * 80%) * 30%. We assume that 20% of the total memory will be used by the OS. This is the starting value. This may need to be adjusted upwards.	MEMORY_TARGET = Total Physical Memory * 75%. We assume that 20% of the total memory will be used by the OS. The Oracle instance should be running on a dedicated machine. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 1048576.
4 GB	SGA_TARGET = (Total Physical Memory * 80%) * 60%. We assume that 20% of the total memory will be used by the OS. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 1048576. PGA_AGGREGATE_TARGET = (Total Physical Memory * 80%) * 30%. We assume that 20% of the total memory will be used by the OS. This is the starting value. This may need to be adjusted upwards.	MEMORY_TARGET = Total Physical Memory * 75%. We assume that 20% of the total memory will be used by the OS. The Oracle instance should be running on a dedicated machine. Statistics level should be TYPICAL or ALL. LOG_BUFFER = 1048576.

Tuning Oracle 10gR2 Databases

This section provides information about tuning Oracle 10g databases.

Automatic Shared Memory Management

Oracle 10g utilizes Automatic Shared Memory Management (ASMM) of individual SGA components like shared pool, java pool, large pool and db cache. You do not need to estimate when setting the size of SGA components. In fact, there is no need to set any parameters defining SGA size.

All you have to do is to set a new parameter called SGA_TARGET. The parameter SGA_TARGET takes a value which indicates the maximum size of SGA required for your instance.

Consider that you set SGA_TARGET to say 800MB. This indicates that maximum size to which SGA can grow is 800MB. All the SGA components like shared pool, buffer cache, large pool, java pool will be allocated from this 800M maximum SGA. Oracle will automatically calculate the initial size of these components and resizes it as per the requirement without any manual intervention.

You do not have to explicitly define values for shared pool, buffer cache, large pool and java pool if you set SGA_TARGET. The SGA_TARGET will be limited by the SGA_MAX_SIZE value. The SGA_MAX_SIZE cannot

be modified dynamically. If SGA_MAX_SIZE is not set, both the parameters have the same value and it will be not possible to increase the size of SGA_TARGET dynamically.

Automatic Segment Space Management

The Automatic Segment Space Management (ASSM) feature allows Oracle to use bitmaps to manage the free space within segments. The bitmap describes the status of each data block within a segment with respect to the amount of space in the block available for inserting rows. The current status of the space available in a data block is reflected in the bitmap allowing Oracle to manage free space automatically with ASSM.

ASSM tablespaces automate freelist management and remove the ability to specify PCTUSED, FREELISTS, and FREELIST GROUPS storage parameters for individual tables and indexes created in ASSM tablespaces. The values for parameters PCTUSED and FREELISTS are ignored and Oracle automatically manages the space for these tables and indexes inside the tablespace using bitmaps. PCTFREE can still be specified and is used with ASSM.

Tuning Oracle 11g Databases

This section provides information about tuning Oracle 11g databases.

Automatic Memory Management (AMM)

Beginning with version 11g, the Oracle database can automatically manage the SGA memory and the instance PGA memory. You only need to designate the total memory size to be used by the instance, and the Oracle database will dynamically exchange memory between the SGA and the instance PGA as needed to meet processing demands. This capability is referred to as automatic memory management. With this memory management method, the database also dynamically tunes the sizes of the individual SGA components and the sizes of the individual PGAs.

AMM is implemented using Memory_Target and Memory_max_target parameters. Use of MEMORY_MAX_TARGET is optional. When MEMORY_MAX_TARGET is not set, Oracle automatically sets MEMORY_MAX_TARGET to the value of MEMORY_TARGET.

To switch to AMM, define the value of MEMORY_TARGET in the spfile. It is important to ensure that the values of SGA_TARGET and PGA_AGGREGATE_TARGET are set to 0. Customers upgrading to 11g instance must set the value of SGA_TARGET and PGA_AGGREGATE_TARGET to 0.

Configuring PostgreSQL Databases

The following information will help you use the PostgreSQL database with Atlas Hub:

- It is installed in the pgsql folder under the Atlas Hub folder.
- The service name is PostgreSQL-9.3_ST.
- The superuser login name is postgres.
- The following Atlas Hub functionality is not available for PostgreSQL: Online Purge, Maintenance Scheduler, and the Import/Export Manager.

When connecting to a PostgreSQL database server on Microsoft Windows, the Microsoft Windows path should include the path to the PostgreSQL bin directory. If the path to the PostgreSQL libraries is not set correctly, a message similar to the following one will appear:

```
Libpq.dll: The specified module could not be found.
pq.dll: The specified module could not be found.
DBMS API Library loading fails. This library is a part of DBMS client
installation, not SQLAPI++.
Make sure DBMS client is installed and this required library is available for
dynamic loading.
```

When connecting to a PostgreSQL database server on Microsoft Windows, the Microsoft Windows system path should include the path to the PostgreSQL bin directory. For example, for a default installation, add this to the System PATH variable: `C:\Program Files\PostgreSQL\9.3\bin`.

Important: When a new database is created, all the default PostgreSQL settings (for example, file location and size) are used except for server encoding. This setting must be UTF-8.

Creating a Server Configuration (for an Existing Database)

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create an Atlas Hub configuration using a previously created PostgreSQL database.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field. If you want the server configuration to have the same name as the database (a nice convention, especially if you have several server configurations), you must follow the database naming conventions explained at the beginning of this section.
   b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
   c) Select **PostgreSQL** from the **Database Type** list.
   d) Uncheck the option to **Create new StarTeam database**, so that Atlas Hub will not automatically create the database for it.
   e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

   f) Click **Next**, and enter the **Database Server name**, **Database name**, **Database login name**, and password in the appropriate text boxes.

g) Optionally, if you are using a port other than the default, check **Edit Database Port** and type the port number in the text field.

h) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.

i) Click **Finish**. This action re-displays the **Server Administration** tool, which shows your new server configuration as a child of the `Local` node.

Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

j) Optionally, if you are using a port other than the default, check **Edit Database Port** and type the port number in the text field.

k) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.

l) Click **Finish**. This action re-displays the **Server Administration** tool, which shows your new server configuration as a child of the `Local` node.

Note: In addition to creating the server configuration, Atlas Hub adds information about the new server configuration to your `starteam-server-configs.xml` file. For more information about this file, see the *Server Administration Tool Help*.

4. By default, all server configurations are set to use the TCP/IP endpoint (port) 49201. However, each server configuration on a given computer must have a unique endpoint so it is recommended that you edit the default endpoint. To change the endpoint:
   a) Select the server configuration.
   b) Click the **Start with Override** button (or click **Actions** > **Start with Override** from the main menu). The **Start with Override** dialog box opens.
   c) Enter the endpoint that you want to use in the **TCP/IP Endpoint** field, and click **OK**.
5. Be sure to configure your new server configuration (for information, see the *Server Administration Tool Help*) and plan a backup schedule for it.

Creating and Starting a Server Configuration

The first time you start a new server configuration, Atlas Hub creates all tables in the database you specify. This section explains how to create a server configuration and start it for the first time. It assumes that you want the server to automatically create a PostgreSQL database. If that is not the case, see *Creating a Database Manually*.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

1. Start the **Server Administration** tool. Click **Start** > **Programs** > **Micro Focus** > **Hub** > **Server Administration**. The **Server Administration** tool opens.
2. Click **Server** > **New Configuration**. The **New Configuration** dialog box opens.
3. Enter the new configuration data:
   a) Type the name of the configuration in the **Configuration Name** field. If you want the server configuration to have the same name as the database (a nice convention, especially if you have several server configurations), you must follow the database naming conventions explained at the beginning of this section.

b) Type or click **Browse** to specify the **Repository Path** location to be used to store log files and other information. If the repository path that you enter does not exist, the application creates it for you. The **Repository Path** is also the location for the default hive.
c) Select **PostgreSQL** from the **Database Type** list.
d) Check the option to **Create new StarTeam database**, so that Atlas Hub automatically creates the database.
e) Create an initial hive for the Native-II vault by doing one of the following:

Accept the default settings	Leave the **Default** option selected and proceed to the next step. With the default settings, Atlas Hub: • Creates an initial hive named `DefaultHive`. • Creates subfolders on the repository path named `Archives` and `Cache` to be used by the `DefaultHive`. • Stipulates that the maximum cache size is 20% of the space currently available on the drive on which the cache is located. • Uses the default setting of 600 seconds (10 minutes) between cache cleanups. • Uses the default setting of 95% for the storage threshold, the point at which this drive is considered full.
Specify custom values	Select the **Custom** option and change any of the hive settings.

f) Click **Next** to create the PostgreSQL database.

4. Enter the server and database information:
   a) Enter the name in the **Database Server name** field.
   b) Type or click **Browse** to specify the names of the computer and the database on your network that should be used.
   c) Enter the password for the system administrator in the **Database Server name** field. The initial default system administrator password is `postgres`.
   d) Click **Verify Connection** to test the connection. If the connection fails, review and change your settings.
   e) Click **Next**.
5. Enter the information for creating the data files and transaction logs, and click **Finish**.

   Note: We recommend keeping the data files and transaction log files on different physical drives under their own disk controllers.

   The default settings are appropriate for your use if you have fewer than 15 users and expect to store 1GB or less data.
6. By default, all server configurations are set to use the TCP/IP endpoint (port) 49201. However, each server configuration on a given computer must have a unique endpoint so it is recommended that you edit the default endpoint. To change the endpoint:
   a) Select the server configuration.
   b) Click the **Start with Override** button (or click **Actions** > **Start with Override** from the main menu). The **Start with Override** dialog box opens.
   c) Enter the endpoint that you want to use in the **TCP/IP Endpoint** field, and click **OK**.
7. Be sure to configure your new server configuration (for information, see the *Server Administration Tool Help*) and plan a backup schedule for it.

SQL Scripts for PostgreSQL Databases

Atlas Hub comes with some SQL scripts written specifically for use with your database. These scripts help you maintain and tune Atlas Hub databases. You run some SQL scripts after installation, some on a weekly basis for database performance maintenance, and some scripts are run for you automatically by Atlas Hub.

The SQL scripts for PostgreSQL databases that you may run are located in the `Micro Focus\Hub \DBScripts\postgresql_Scripts` folder.

`starteam_postgresql_dropall.sql`

Run: only if necessary.

Caution: Running this script deletes all Atlas Hub tables and the data they contain from the database. Use this script with extreme caution.

One use case example is if you migrate an Atlas Hub configuration to another database, you might use this script to remove tables from the original database. Another example is if you mistakenly add the Atlas Hub tables to a tablespace other than the Atlas Hub tablespace, use this script to remove them.

Creating a Database Manually

Despite the fact that Atlas Hub has automated PostgreSQL database creation, you may prefer to create your own. This makes more sense for PostgreSQL because there are good tools for database creation.

Database names should:

- Begin with a letter.
- Contain letters and numbers only.
- Not contain spaces.
- Not be a SQL reserved word such as `create`, `delete`, `if`, `then`, `else`, or `goto`.

Important: The **Server Administration database** options may fail to run for databases with names that do not follow these guidelines.

To create your own PostgreSQL Server database:

1. Install PostgreSQL Server.
2. Install Atlas Hub.
3. Create an Atlas Hub database. Contact your database administrator about the specifics. Be sure that:
   - The database is owned by an Atlas Hub user.
   - The name of the database follows the conventions explained earlier in this section.
   - The template for the database is template0.
   - The encoding for the database should be: `UTF-8.` This refers to the physical storage of character strings in the database.
4. Create and start a server configuration. See *Creating a Server Configuration (for an Existing Database)*

Backups

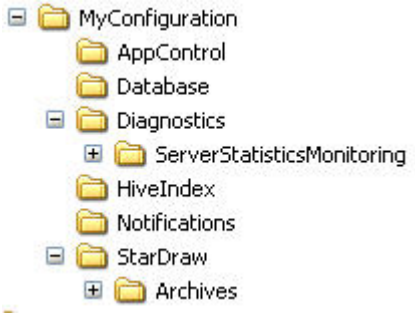
We recommend using the **pg_dump** utility for backing up a PostgreSQL database. It makes consistent backups even if the database is being used concurrently. **pg_dump** does not block other users accessing the database (readers or writers). The most flexible output file formats is the `directory format` (`-Fd`). This format is compressed by default.

Data Locations and Repositories

As part of creating a new server configuration, Atlas Hub creates a number of folders for storing log files, archive files, and so on. This section explains the location and purpose of the files and folders that make up the Atlas Hub vaults and repositories.

Repositories

The following figure shows the server configuration whose repository path starts with a drive letter (not shown in figure) and ends with the folder name `MyConfiguration`.



The name of the server configuration may also be `MyConfiguration`.

The repository path is a general location for initial storage of a variety of objects, most of which can be moved to new locations later, independent of one another.

Native-II Vault Folders

For server configurations, the repository path is also the initial home of several folders used by the Native-II Vault to store archive files and information about them:

Default Hive

If you accepted all the defaults when you created the server configuration or if you started an upgraded server configuration without first creating a hive, Atlas Hub automatically creates the folder `DefaultHive`. It is a subfolder of the repository path and is created when you start the server configuration for the first time.

Whether the initial hive is called `DefaultHive` or not, you will have at least one hive for each server configuration. You may have several hives. Each hive has an archive and cache path. An easy, but not mandatory, naming convention is the one illustrated in the above figure. The name of the hive becomes the name of a folder with two subfolders: `Archives` and `Cache`. However, you can place these paths anywhere. They do not need to be on the same drive or volume.

The `DefaultHive` subfolders have the following purposes:

Archives This folder stores file revisions in archive files, which may be compressed.

Cache This folder stores uncompressed versions of archive files. It has two subfolders `Temp` and `Deltas`. Temp is used for files that are being added to Atlas Hub and for new file revisions that are being checked in. `Deltas` stores the differences between working files and tip revisions when a user asks that transmissions over slow connections be optimized.

You can use the **Hive Manager** to change an individual hive's archive path or cache path. Such changes should be done only when that hive must be moved. For example, you might move a hive as a result of a drive failure. You would also need to copy the contents of the hive's archive path to the new location.

The archive path for any hive must be included in a backup. A hive's cache path does not need to be included.

HiveIndex If you started an Atlas Hub configuration, it has at least one hive. The `HiveIndex` folder stores the `hive-index.xml` file, which contains the properties for each hive used by the server configuration.

You can change the path to the `HiveIndex` folder by changing the repository path in the `starteam-server-configs.xml` file. You would make this change only when necessary, for example, because of a drive failure.

The `HiveIndex` folder must be included in a backup.

Archive and Cache Path Structure

Every hive's archive path and cache path has the same structure. This structure is similar to that used by Atlas Hub clients to store file status records. The files located in the archive and cache are organized into subfolders. This makes browsing and managing the files easier. The name of the subfolders in which a file revision is stored is based on the initial characters in the archive file's name. For example, suppose a file revision's contents has an MD5 value of `01fc3c4ac5e0e92cc707f30fb73a0726`. Assuming the user specified an archive path of `C:\DefaultHive\Archives` the archives path for this revision would be one of the following, depending on whether or not the archive file was compressed:

```
C:\DefaultHive\Archives\01\f\01fc3c4ac5e0e92cc707f30fb73a0726
C:\DefaultHive\Archives\01\f\01fc3c4ac5e0e92cc707f30fb73a0726.gz
```

The archive path for each hive (for example `C:\DefaultHive\Archives`) must be included in a backup.

Log Files and Folders

The repository path folder becomes the home of the following related objects:

Server log files A new server log file is created each time you start the server configuration. See the **Server Administration** tool help for more information.

.dmp files These are any .dmp files created when you use server diagnostics to create `.DMP` files for errors and unexpected conditions encountered by the server. Usually, you have no `.DMP` files or trace files unless a technical support representative is working with you on a problem. See the **Server Administration** tool help for more information.

Notifications folder When an Atlas Hub configuration starts for the first time, the contents of the `Notifications` folder in the installation directory are copied to the repository for the server configuration in a corresponding `Notifications` folder.

Trace folder The `Trace` folder stores the files that are created when and if you use server diagnostics to trace server commands. See the **Server Administration** tool help for more information.

These objects do not have to remain in the repository path. You can change the path to all of the above by changing the `Log Path` using the **Server Administration** tool.

These folders do not have to be included in a backup.

Atlas Hub Backups

Atlas Hub allows you to perform backups in a completely online manner, without shutting down or locking a server configuration.

What to Backup

When performing a backup, you need to include all of the following:

Application Configuration File	The `starteamserver-configs.xml` file needs to be backed up. For Atlas Hub, it is located here: `C:\Program Files\Micro Focus\Hub\`
Database Files	Backup the database. Each server configuration has one database (or, in the case of Oracle, one schema user).
Server Configuration Directory	In the `starteamserver-configs.xml`, locate the `value` of the `RepositoryPath` option. Include everything in that directory. For example: `<option name="RepositoryPath" value="c:\my\server-config\directory-name"/>` In the example, back up everything in `c:\my\server-config\directory-name`.

All of these should be backed up at the same time, preferably on the same media.

Important: The server configuration directory should not be backed up until after the database backup completes. These two backups should not be done simultaneously. This guarantees that everything referenced in the database appears in the server configuration in the right data location.

Restoring Data

There may be a few scenarios when you need to restore data. In the event that:

The database is lost	The administrator must restore the last full backup of the database and apply the redo logs (Oracle) or apply incremental backups (Microsoft SQL Server) to roll forward the database to the vault time.
The vault is lost	It is very important to take a backup of the database in its current state, including the transaction logs (redo logs), before performing any restoring.
The vault (or both the vault and database) are lost	The server administrator must restore both the database and the vault from the last backup. After restoring the online database backup, the database has to be rolled forward to the vault backup time.

Note: If you do not have transaction logs (redo logs) available, this can cause data loss and limit your disaster recovery capabilities.

1. Reload the database from the last backup.
2. Simultaneously, reload the repository information (server configuration directory) from the last backup. If the full and incremental backups were used, then you can reload the last full backup, and in parallel, reload all subsequent incremental backups.
3. Restore the `starteam-server-configs.xml` file.
4. Start the server configuration.
5. Test the server configuration.

When all loads are complete, the repository is ready to use. It is okay if the `archive` or `attachment` folders have “future” files not represented in the database. Atlas Hub ignores the “future” files and, if those file revisions are eventually added again, Atlas Hub overwrites the existing files.

Database Backups Overview

This section outlines the backup options available to DBAs and makes recommendations for backing up the databases used by the server configurations. Be aware that these are just recommendations. Any finalized disaster recovery plan must be created by your organization in consultation with its IT infrastructure staff.

Note: An application backup consists of backing up both the database and the application archive files.

Microsoft SQL Server Database Backups

For server configuration online backups, it is essential to take full database and transaction log backups.

Microsoft SQL Server Backup Types and Recovery Models

Microsoft SQL Server supports the following types of backups:

Full database backup	A full database backup contains the full copy of the database as it was at the time when the backup was initiated. Full backups provide a snapshot of the database. Most of the recovery options require a full backup to be available.
Differential backup	A differential database backup records only the data that has changed since the last full database backup. Scheduling frequent differential backups is a good idea because the backups are smaller and they complete quickly. A differential backup without a prior full backup is useless.
Transaction log backup	A transaction log backup includes all the transactions since the last transaction log backup. Transaction log backups enable recovery up to the last committed transaction.
A file or file group backup	A file or file group backup consists of backing up individual data files (or the files in the file group). The files in a database can be backed up and restored individually.

The entire database can be recreated from a database backup in one step by restoring the database. The restore process overwrites the existing database or creates the database if it does not exist. The restored database will match the state of the database at the time the backup completed, minus any uncommitted transactions. Uncommitted transactions are rolled back when the database is recovered.

Based on the resource requirements, the DBA can also choose the recovery model for the database. The recovery model balances logging overhead against the criticality of fully recovering the data.

The recovery models supported by Microsoft SQL Server are:

Full	The data is critical and must be recoverable to the point of failure. All data modifications are logged. All Microsoft SQL Server recovery options are available.
Bulk-logged	Certain bulk operations, such as bulk copy operations, `SELECT INTO`, and text processing, can be replayed if necessary, so these operations are not fully logged. You can recover only to the end the last database or log backup.
Simple	All data modifications made since the last backup are not available. This type of recovery scenario has the lowest logging overhead, but cannot recover past the end of the last backup.

Recovering to a point-in-time (for example, a time before unwanted data was entered) requires either full or bulk-logged recovery models.

Microsoft SQL Server Full Database Backups

A full database backup creates a duplicate of the data that is in the database. This is a single operation, usually scheduled at regular intervals. Full database backups are self-contained. Full backups provide a snapshot of the database. Most of the recovery options require a full backup to be present.

We strongly recommend the use of full backups.

Microsoft SQL Server Differential Database Backups

A differential database backup records only the data that has changed since the last database backup. Frequent differential backups are recommended to reduce backup times. Making frequent backups decreases the risk of losing data.

Differential backups restore the data that they contain to the database. Differential backups cannot be used to recover the database to a point in time.

The availability of a differential backup minimizes the time it takes to roll forward transaction log backups when restoring a database.

Microsoft SQL Server Transaction Log Backups

The transaction log is a serial record of all the transactions that have been performed against the database since the transaction log was last backed up. With transaction log backups, you can recover the database to a specific point in time or to the point of failure.

When restoring a transaction log backup, Microsoft SQL Server rolls forward all the changes recorded in the transaction log. When Microsoft SQL Server reaches the end of the transaction log, it has recreated the exact state of the database at the time of the backup operation.

If the database is recovered, Microsoft SQL Server then rolls back all transactions that were incomplete when the backup operation started.

Transaction log backups generally use fewer resources than database backups. As a result, you can create them more frequently than database backups. Frequent backups decrease the risk of losing data. For high volume Online Transaction Processing (OLTP) environments, it is desirable to create transaction log backups more frequently. Transaction log backups can only be used with Full and bulk-logged recovery models.

The transaction log cannot be backed up during a full database backup or a differential database backup. However, the transaction log can be backed up while a file backup is running.

Never backup a transaction log before a database backup is created because the transaction log contains the changes made to the database after the last backup was created.

Never truncate the transaction log manually because it breaks the backup chain. If a transaction log has been truncated, take a full database backup to start a new backup chain.

Microsoft SQL Server File Backups

A file or file group backup consists of the backing up of individual data files (or the files in the file group). A file-based recovery model increases the speed of recovery by allowing you to restore only the damaged files without restoring the rest of the database. For example, suppose a database is comprised of several files located physically on different disks and one disk fails. Only the file on the failed disk needs to be restored and rebuilt using the transaction log backup.

File backup and restore operations must be used in conjunction with transaction log backups. For this reason, file backups can only be used with the full recovery and bulk-logged recovery models.

Microsoft SQL Server Database Backup Recommendations

We recommend that you:

- Use the full recovery model.
- Perform a full database backup once every day. For full database sizes greater than 3 GB, it is okay to perform full backups on alternate days. If you perform full backups on alternate days, we strongly recommend that you create daily differential backups.
- Create daily transaction log backups after the completion of the full or differential backup. In addition to this, schedule a transaction log backup every 4 hours. Never truncate a transaction log manually.
- In case of a disaster, create a backup of the currently active transaction log. If active transaction log backup is not available (for example, because a media failure for the drive hosting the transaction logs

and drive is not being mirrored), the database cannot be recovered past the last available transaction log backup. This would hamper a point-in-time recovery beyond the last available transaction log backup.

Oracle Database Backups

An online or hot backup is a backup performed while the database is online and available for read/write operations. Except for Oracle exports, you can only perform online backups when running in `ARCHIVELOG` mode. An offline or cold backup is a backup performed while the database is offline and unavailable to its users.

The remainder of this topic explains the types of backups supported by Oracle and provides recommendations about performing Oracle backups.

Oracle Backup Types

Typically an Oracle DBA uses one or more of the following options to back up an Oracle database.

Export/Import	Exports are *logical* database backups that extract logical definitions and data from the database to a file. Export backups are cross-platform and can be easily moved from one operating system to the other.
Cold or Offline Backups	These backups require shutting down the database instance and copying all the data, log, and control files.
Hot or Online Backups	These backups are taken when the database is available and running in `ARCHIVELOG` mode. To perform a backup of this type, the tablespaces need to be in backup mode and all the data files associated with the tablespace must be backed up. It is essential to backup the control files and archived redo log files.
RMAN Backups	While the database is offline or online, DBAs can use the RMAN utility to back up the database.
Export/Import Data Pump	Export pump and import pump are new for Oracle 10g. `Expdp` and `Impdp` are cross-platform and can be easily moved from one operating system to the other.

Oracle Logical Database Backups

Oracle exports are *logical* database backups (not physical) as they extract data and logical definitions from the database into a file. Other backup strategies normally back up the physical data files. One of the advantages of exports is that you can selectively re-import tables. However, you cannot roll forward from a restored export file. To completely restore a database from an export file, you almost need to recreate the entire database. Logical backups takes a snapshot of the database schema as it was at a particular time.

Oracle Offline/Cold Database Backups

A backup performed when the database is shut down is known as an offline or cold backup. You must copy the data files, control file and online redo log files using an OS copy utility. This is a considered a complete backup of the database. Any changes made after this backup are unrecoverable if the database is running in `NOARCHIVELOG` mode. All transactions are recorded in online redo log files whether the database is archiving or not. When redo logs are archived (`ARCHIVELOG` mode), Oracle allows you to apply these transactions after restoring files that were damaged (assuming that an active redo log file was not among the files damaged).

Whenever the schema of the database is changed, such as when you add a new data file, rename a file, or create or drop a tablespace is created, you must shut down the database and copy at least the control file and the newly added data file. A complete backup of the database is preferred.

Before performing a cold backup, it is essential to get a list of all the Oracle files that need to be backed up. Running the following queries will provide a list of all the files.

```
select name from sys.v_$datafile;
select member from sys.v_$logfile;
select name from sys.v_$controlfile;
```

Shut down the database from SQL*Plus or Server Manager. Back up all the files to secondary storage (for example, tapes). Ensure that you back up all data files, all control files, and all log files. When completed, restart your database.

Note: If your database is in `ARCHIVELOG` mode, you can still use archived log files to roll forward from an offline backup. If you cannot take your database down for an offline backup at a convenient time, switch your database into `ARCHIVELOG` mode and perform an online backups.

Oracle Online/Hot Database Backups

A backup performed when the database instance is running is known as online or hot backup. Online backups are very important at customer sites where a database instance must operate 24-hours per day and offline backups are not feasible. During the duration of an online backup, the database remains available for both reading and updating. For this kind of backup, the database must be in ARCHIVELOG mode. Only data files and current control file need to be backed up. Unlike offline backups, the unit of a online backup is a tablespace, and any or all tablespaces can be backed up whenever needed. Different data files can be backed up at different times.

To perform an online backup, you switch the tablespace into “backup mode” before copying the files as shown in the following example.

```
ALTER TABLESPACE xyz BEGIN BACKUP;
! cp xyfFile1 /backupDir/
ALTER TABLESPACE xyz END BACKUP;
```

It is better to backup individual tablespaces than to put all tablespaces in backup mode at the same time. Backing them up separately incurs less overhead. After completing the tablespace backups, it is important to back up the control files as shown in the following example.

```
ALTER SYSTEM SWITCH LOGFILE; -- Force log switch to update control file
headers
ALTER DATABASE BACKUP CONTROLFILE TO '/<directory name>/control.dbf';
```

The frequency of online backups is inversely proportional to the time taken for recovery in case of a media failure. The older your backup, the more redo log files need to be applied, and the recovery times increases. Backup strategies should be tested before being used to protect a production database.

We strongly recommend that you run online backups at times when the database is least accessed, during non-peak hours. Oracle writes complete database blocks instead of the normal deltas to redo log files while in backup mode. This leads to excessive database archiving and even database freezes.

Oracle RMAN Database Backups

Recovery Manager (RMAN) is an Oracle tool that lets the DBA back up and recover Oracle databases. RMAN lets you perform full backups (with the database online or offline), incremental backups on the block level, and backups of online redo logs and control files.

The SYSDBA privilege is required to run RMAN on a database. The other benefits of RMAN backups are that you can:

- Keep track of all backup and recovery operations performed against the database.
- Manage centralized backup and recovery procedures for the enterprise.
- Identify corrupt blocks.
- Back up only those blocks that actually contain data. This can lead to significant savings in backup space requirements.

- Have support for incremental backups. Incremental backups back up only those blocks that have changed since a previous backup. This helps with the disk space usage and reduces the backup times significantly. Oracle 10g has introduced a new feature called “block change training”. This feature provides significant improvement for incremental backups. Contact your DBA about how to implement this feature.

The following examples of RMAN backup and restore are extremely simplistic and are included on to illustrate basic concepts. By default, Oracle uses the database control files to store information about backups. Normally, you will prefer to set up an RMAN catalog database to store RMAN metadata. Read the *Oracle Backup and Recovery Guide* before implementing any RMAN backups.

```
rman target sys/*** nocatalog
 run {
  allocate channel t1 type disk;
  backup format '/app/oracle/db_backup/%d_t%t_s%s_p%p' (database);
  release channel t1;
 }
Example RMAN restore:
 rman target sys/*** nocatalog
 run {
  allocate channel t1 type disk;
  restore tablespace users;
  recover tablespace users;
  release channel t1;
 }
```

Oracle Export/Import Data Pump

Oracle introduced the export/import data pump in the 10g release. The import pump is twenty times faster than the conventional import utility. Export/Import data pump utilities are *logical* database backups (not physical) as they extract data and logical definitions from the database into a file. Export/Import data pump utilities do not fit into 24/7 model because they do not offer roll-forward capabilities. Export data pump provides a snapshot of the database schema as it was at a particular time.

Oracle Database Backup Recommendations

We strongly recommend the use of RMAN backups if your enterprise wants to run an Atlas Hub instance in a 24/7 environment. RMAN has evolved over the last few years and Oracle Corporation continues to add features that make disaster recovery easier, more reliable, and faster.

PostgreSQL Database Backups

Backups can be one of the most critical aspects of administering any database, and PostgreSQL is no exception. While the PostgreSQL database is very versatile and resilient, problems can happen. A power failure could occur corrupting the database or the hard-drive could fail. You could also have problems with users, authorized or unauthorized, changing or destroying data.

PostgreSQL database provides easy backup and restore utilities. There are two different types of backups you can use:

Important: The following information is provided to point you in the right direction only. Please refer to the PostgreSQL database documentation for full procedures.

SQL text backups SQL text backups allow you to backup your entire server including user and passwords but you cannot restore a single database or just the users unless you edit the file. Both of these files are created using the same `pg_dump` command depending on the options you specify. Both are restored using different commands. Another thing to consider is that binary backups can be restored using `pgAdmin`.

Binary backups Binary backups are convenient and let you do a couple different things when you re-load them, however, they are limited to backing up a single database and do not include users and passwords.

Use the following commands to manage your PostgreSQL databases.

Backup
```
pg_dump -U postgres -h <hostname> -p 5432 -F c -f Atlas.backup
<Database Name>
```

- `<Hostname>` is the name of the database machine.
- `<Database Name>` is the name of the ST database.
- `-F c` would create a compressed binary backup

Create
```
createdb -U postgres -h <New Hostname> -p 5432 <New Db Name>
```

Restore
```
pg_restore -U postgres -h <New Hostname> -p 5432 -d <New Db Name>
Atlas.backup
```

Enabling SSL

SSL, or Secure Socket Layer, is a technology which allows web browsers and web servers to communicate over a secure connection. This means that the data being sent is encrypted by one side, transmitted, and then decrypted by the other side before processing. This is a two-way process, meaning that both the server and the browser encrypt all traffic before sending out data.

Another important aspect of the SSL protocol is *authentication*. This means that during your initial attempt to communicate with a web server over a secure connection, that server will present your web browser with a set of credentials, in the form of a *certificate*, as proof the site is who and what it claims to be. Therefore to implement SSL, a web server must have an associated certificate for each external interface (IP address) that accepts secure connections. The theory behind this design is that a server should provide some kind of reasonable assurance that its owner is who you think it is, particularly before receiving any sensitive information.

Installing the SSL Certificate

1. To import the root certificate:
   ```
   keytool -import -alias root -keystore [path/to/your/keystore]
    -trustcacerts -file [path/to/the/root_certificate]
   ```
2. To import your new certificate:
   ```
   keytool -import -alias [youralias] -keystore
   [path/to/your/keystore] -file [path/to/your_keystore]
   ```

Configuring Tomcat to use SSL Connectors

1. Open `$CATALINA_BASE/conf/server.xml`. This is Tomcat's main configuration file that contains the global connector options.
2. Search for port `8443`.

   It will look like this:
   ```
   <!-- Define a SSL HTTP/1.1 Connector on port 8443
   ```

This connector uses the JSSE configuration, when using APR, the connector should be using the OpenSSL style configuration described in the APR documentation.

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
maxThreads="150" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"/>
```

You'll notice that the comment enclosing this connector talks about a choice between APR and JSSE configurations. This refers to the implementation of SSL you are intending to use. JSSE, which is Tomcat's default configuration, is supported by default, and included in all JDKs after version 1.4. So if you don't even know what APR is, you only need to uncomment this entry, and add some additional information to allow Tomcat to find your keystore.

3. Add the following:

```
<Connector port="8443" maxThreads="150" scheme="https"
secure="true" SSLEnabled="true" keystoreFile="path/to/your/keystore"
keystorePass="YourKeystorePassword" clientAuth="false"
keyAlias="yourAlias" sslProtocol="TLS"/>
```

4. If, on the other hand, you know that using the Apache Portable Runtime (APR), also known as Tomcat's "native library," is by far the best practice to follow, especially when using Tomcat as a standalone web server (which you probably are), and have already installed it on your server, then you'll need to alter this entry as follows to allow Tomcat to use APR's OpenSSL implementation in place of JSSE, or trying to use SSL will generate an error:

```
<Connector port="8443" scheme="https" secure="true"
SSLEnabled="true" SSLCertificateFile="/path/to/your/certificate.crt"
SSLCertificateKeyFile="/path/to/your/keyfile"
SSLPassword="YourKeystorePassword"
SSLCertificateChainFile="path/to/your/root/certificate"
keyAlias="yourAlias" SSLProtocol="TLSv1"/>
```

Notice that if you are using APR, the `SSLCertificateFile` and `SSLCertificateKey-type` attributes are used in place of the `keystoreFile` attribute. For more information on the differences between using APR in place of JSSE, refer to Apache's Tomcat APR Documentation.

5. Restart Tomcat.
6. Once you're up and running again, test your configuration by connecting to a secure page, using a URL such as `https:/[yourhost]:8443`. If you followed the directions correctly, you should be able to view the page over a secure HTTPS connection

Ordering an SSL certificate for production

1. Generate a private key or use an existing one, by which you generate a certificate signing request.
2. The CSR file is then sent over to a trusted signing authority who can issue a trusted public certificate.
3. The web-server that accepts secure connections must then be configured to use the certificate.

Generate a Certificate Signing Request (CSR)

In order to obtain a certificate from the certificate authority of your choice you have to create a Certificate Signing Request (CSR). That CSR will be used by the certificate authority to create a certificate that will identify your website as "secure". To create a CSR follow these steps:

1. Create a local self-signed certificate.
2. The CSR is then created using the following command:

```
openssl req –out {output-csr-file} –key {private-key-file} -new
```

Importing the certificate

The signing authority would send you two files, the *actual certificate* and *chain certificates*.

Importing the certificate into JAVA keystore

1. Use the following command to import the chain certificate:
```
keytool -import -alias root -keystore <your_keystore_filename>
    -trustcacerts -file <filename_of_the_chain_certificate>
```
2. Use the following command to import the public certificate:
```
keytool -import -alias tomcat -keystore <your_keystore_filename>
    -file <your_certificate_filename>
```

Importing the certificate for AWS beanstalk

You need to update an SSL certificate for an HTTPS load balancer:

1. Open the Amazon EC2 console at *https://console.aws.amazon.com/ec2/*.
2. In the navigation pane, under **LOAD BALANCING**, click **Load Balancers**.
3. Select your load balancer.
4. In the **Listeners** tab, click **Change** in the **SSL Certificate** column for the certificate.
5. In the **Select Certificate** dialog box, do one of the following:
   - If you have already uploaded an SSL certificate using IAM, select **Choose an existing SSL Certificates**, select the certificate from **Certificate Name**, and then click **Save**.
   - If you have an SSL certificate to upload, select **Upload a new SSL Certificate**. Enter a name for the certificate, copy the required information to the form, and then click **Save**. Note that the certificate chain is not required if the certificate is a self-signed certificate.

Troubleshooting

Debugging SSL certificate connection issues with a web-server

Open SSL provides a utility to test the SSL handshake and check what certificates are returned by a web-server. You can use the following command:

```
openssl s_client –connect cas.borland.com:443
```

Testing SSL connection for a server

A simple Java program can test the SSL connection:

```
package com.microfocus.cloud.audit;

import javax.net.ssl.SSLSocket;
import javax.net.ssl.SSLSocketFactory;
import java.io.*;

/** Establish an SSL connection to a host and port, writes a byte and
 * prints the response. See
 * http://confluence.atlassian.com/display/JIRA/Connecting+to+SSL+services
 */
public class SSLPoke {
    public static void main(String[] args) {
        if (args.length != 2) {
            System.out.println("Usage: "+SSLPoke.class.getName()+" <host>
<port>");
            System.exit(1);
        }
        try {
            SSLSocketFactory sslsocketfactory = (SSLSocketFactory)
SSLSocketFactory.getDefault();
            SSLSocket sslsocket = (SSLSocket)
sslsocketfactory.createSocket(args[0], Integer.parseInt(args[1]));
```

```
            InputStream in = sslsocket.getInputStream();
            OutputStream out = sslsocket.getOutputStream();

            // Write a test byte to get a reaction :)
            out.write(1);

            while (in.available() > 0) {
                System.out.print(in.read());
            }
            System.out.println("Successfully connected");

        } catch (Exception exception) {
            exception.printStackTrace();
        }
    }
}
```

Connectors

Micro Focus Connect is a tool used to synchronize item and relationship data between various Micro Focus software change and configuration management systems and other repositories or clients. Once data is synchronized into the system, all of the change management capabilities of the system --including versioning, branching, tracing, labeling, and reporting-- are available on that data.

Micro Focus Connect has a user guide that will help you get started with the product. Additionally, each connector that you use will have a `Readme.html` file that describes required connection information and any issues or limitations.

Configuring the Rally Connector

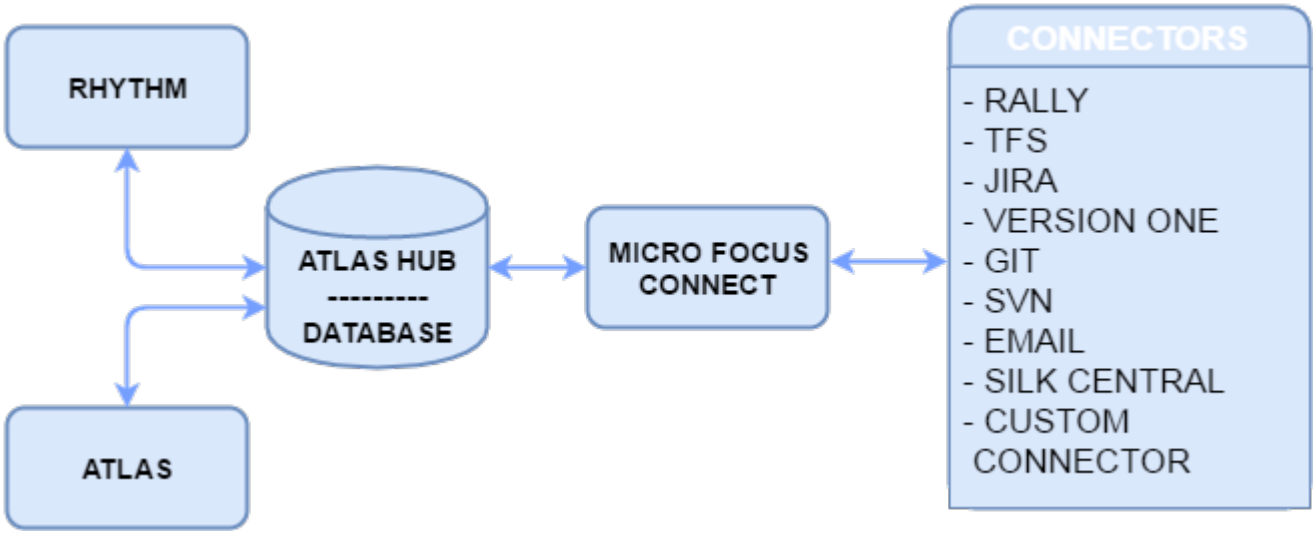
Administrators can use Micro Focus Connect to synchronize assets between Rally and Micro Focus Atlas. To do this, launch Micro Focus Connect and create a new connection to Rally. Follow the procedures in *Using Micro Focus Connect* as well as the *Readme for Rally Connector for Micro Focus Connect*. These resources are installed with Micro Focus Connect.

Tip: If you prefer, you can watch a video on how to configure the connector at *https://www.youtube.com/watch?v=0V0zdS6UXks*.

Configuring the E-mail Connector

Micro Focus Atlas uses Micro Focus Connect to synchronize with your email system. Use the following information to configure the email connector for the synchronization. Refer to the Micro Focus Community and *Using Micro Focus Connect* for further configuration instructions.

Tip: If you prefer, you can watch a video on how to configure the connector at *https://www.youtube.com/watch?v=q8hoK3Z8Wv0*.



Note: The sample `Connect.xml` configuration provided with the email connector demonstrates how to set it up to synchronize Email messages into the Atlas Hub.

Connection Information

Collect and enter the following information in Micro Focus Connect:

Name	Enter a unique name for the data source.
Product	Select the data source product name.
TimeOffset	Number of seconds the data source time differs from the Micro Focus Connect process time.
User	Enter the email address for the user.
Password	Enter the password for the user.
Address	Enter the email address to connect to. For example: `pop.gmail.com`.
Port	Enter the port of the email system.
WatchFolder	Enter the folder to connect to. For example: `Inbox`.
StoreType	Enter the email system store type. For example: `pop3s`. The values can be: `pop3`, `pop3s`, `imap`, or `imaps`. Use the "s" version when a secure port is used by the email server.

Supported Properties

The email connector supports the below properties for the *Email* type (which is associated with the *Concept* type in Atlas Hub):

Subject	Subject text of the email.
From	Sender's address.
Attachments	Returns an HTML string that contains the text with the embedded attachments. The attachments property should be mapped to a *Content* property in Atlas Hub. If the email contains an attachment, and that attachment is an image, then valid html content will be constructed containing that image and assigned to the *Content* property. If the content of the attachment is text then it will be inserted into the *Content* property. If no attachment is included, then the value for the *Content* property will be the body of the email with any embedded images.

Body	Returns a plain text string that contains just the text.
Date	The date the message was sent.
Whiteboard	A link to a *Whiteboard* item whose name property is `Inbox`. The *Whiteboard* property exists to create an Atlas Hub trace between the *Concept* and the *Whiteboard* item.

Project Map Syntax

The project map syntax for the email connector is `address:password`. For example, `john@foo.com:foopassword`.

Gmail Configuration

When using the email Connector with Gmail, you need to enable the **Allow less secure apps** setting in Gmail.

1. Login to your Gmail account.
2. In the **Sign in & Security** section, click **Connected Apps & sites**.
3. Toggle the **Allow less secure apps** setting to `ON`.

Other Connectors

For directions on configuring connectors, visit the Micro Focus Connect Community or refer to the Micro Focus Connect Readme.

Configuring Port Numbers

Atlas uses two Tomcat web servers for its application configuration. The default port values will work in common situations, but you can change them if you need to.

Remember: Each web server that runs on the same machine must have a different port number.

a file called `server.xml`. Each Tomcat web server has its own `server.xml` file.

To change the port number:

1. Open the `server.xml` file for the proper application:

Atlas	`C:\Program Files\Borland\Atlas Planning and Tracking Suite \WebServer\conf`
Atlas Hub	`C:\Program Files\Micro Focus\Hub\WebServer\conf`

2. Locate the following section:

```
<Connector executor="tomcatThreadPool"
               port="80" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8444" />
```

3. Change the port number to a different value.
4. Restart the web server service for this change to take effect.

Configuring E-mail Notifications

In order for users to receive email notifications when they are mentioned in a discussion, an administrator must do the following:

1. On the Atlas Hub installation server, navigate to `C:\Program Files\Micro Focus\Hub \starteamserver-configs.xml`.
2. Open the `starteam-server-configs.xml` file in a text editor.
3. Add `<option name="ALMServiceURL" value="hostname:portnumber"/>` to the file.

   Important: Ensure that you specify an external facing host name rather than `localhost` or `127.0.0.1`.
4. Save and close the file.
5. Open your server configuration in the Atlas Hub **Server Administration** tool.
6. Check **Enable e-mail support**.
7. Type your SMTP server information and the TCP/IP endpoint in the appropriate fields.
8. Check **Enable e-mail notification**.
9. Restart your server.
10. In Atlas, add email addresses to user profiles.

    Note: Email notification will not work unless users have added their email address to their profiles in Atlas.

    Each user must log into their account before the changes take effect.

Configuring the Session Timeout

To change the default session timeout:

1. Open the `ALMConfiguration.xml` file located at `C:\Program Files\Borland\Atlas Planning and Tracking Suite\WebServer\shared\lib\`.
2. Find the `<inactivity>` element.

```
<timeouts>
   <session>
      <!-- Session inactivity mins before session is closed -->
      <inactivity>60</inactivity>
      <!-- Session maximum mins before session is closed -->
      <maximum>600</maximum>
```

3. Change the value of inactivity, in minutes.
4. Save and close the file.

Configuring the Cache Time Out

If you are experiencing sluggishness on the server after long periods of inactivity, you may want to modify the size of the `<cacheTimeOut>` value in the server configuration file.

If you increase the delay, the memory usage on the server will be higher because artifacts remain in memory longer and you will experience quicker loading times. However, the memory will continue to be consumed until your timeout value is reached.

If you decrease the delay, memory usage is better optimized, but you may experience slow response times after long periods of inactivity as it takes longer to load artifacts.

So, you may need to do perform a *balancing act* of adjusting the `cacheTimeOut` value in the server configuration file to find the value that best suits your organization. To do this:

1. Open the `ALMConfiguration.xml` file located at `C:\Program Files\Borland\Atlas Planning and Tracking Suite\WebServer\shared\lib\`.

2. Navigate to the **<configuration>** > **<administration>** node.
3. Find the `<cacheTimeOut>` element. It may be commented out.
4. The default value is `30` hours. Change this to another integer value. If you enter `0`, it means that the server will never release the memory.
5. Save and close the file.

Connecting Atlas to the Atlas Hub

Atlas uses the Atlas Hub to manage its data. Atlas Hub contains *server configurations* that specify the database (among other things) that you will use. You will use the **ConfigManager Utility** to connect Atlas to an Atlas Hub server configuration.

To connect Atlas to an Atlas Hub *Server Configuration*:

1. Open a command prompt with administrator rights.
2. Change directory to: `C:\Program Files\Borland\Atlas Planning and Tracking Suite\`.
3. Run `runConfigManager.bat`.
4. Atlas can only contain connection information to one Atlas Hub. If connection information exists to an existing Atlas Hub configuration, or if you have changed the administrator credentials in the Atlas Hub, you must first delete that connection. Select `D)elete` and then specify the connection to delete.
5. Select `A)dd` to create a connection to a configuration.
6. Enter the connection details for the server configuration:
   - `Server Name`
   - `GUID`
   - `Host Name`
   - `Host Port`
   - `User Name`
   - `Password`
   - `Re-enter Password`
7. Select `Q)uit` to exit the utility. Atlas is now connected to the Atlas Hub configuration that you specified.

Changing the Administrator Credentials

Any time you change the administrator credentials in the Atlas Hub - which is recommended post-installation - you need to change the connection information with the **ConfigManager Utility**. The only way to do that is to delete a connection and then add a new connection with the updated credentials. See *Connecting Atlas to the Atlas Hub*

Verification

After installation, you will want to verify that the applications are accessible and that all services are started.

Services

The following services are required to run Micro Focus Atlas with full capabilities. Verify that these services are started.

- `AtlasSuiteWebServer`
- `BorlandConnect`
- `StarTeamSearchWebService`

Applications

Atlas

Use this URL to log in to Micro Focus Atlas: `http://<machine name>/atlas`.

Important: Default login values are `Administrator` and `Administrator`.

Rhythm

Use this URL to log in to Rhythm: `http://<machine name>/atlas/#perspective=agile`.

Important: Default login values are `Administrator` and `Administrator`.

Micro Focus Connect

Administrators use this URL to login to Micro Focus Connect: `http://<machine name>/ConnectWeb`.

Important: Default login values are `Administrator` and `Administrator`.

Learn more about Micro Focus Connect at *http://community.microfocus.com/borland/managetrack/borland_connect/*.

Atlas Hub Server Admin Tool

Manage your database/server configurations at: `c:\Program Files\Micro Focus\Hub\ServerAdministration.`

Important: Default login values are `Administrator` and `Administrator`.

Index

R

S

U

W