Charles University in Prague
Faculty of Mathematics and Physics

SOFTWARE PROJECT

‹XRouter›

4. Administrators's Guide

Soběslav Benda
Miroslav Cicko
Tomáš Kroupa
Petr Sobotka
Bohumír Zámečník

Supervisor: Mgr. Martin Nečaský, Ph.D.

2011

XRouter – Administrator's guide

Indended audience: administrators who install, configure and run the system
Recommended reading: DaemonNT, XRouter - Definition

Deployment overview

XRouter comes in form of an installer and a database image. The files needed to install an the XRouter system can be obtained freely from the project's download page: http://www.assembla.com/spaces/xrouter/documents/tag/release.

The installer contains an image of an empty database structure, however there can be some updates at the download page.

Requirements

Make sure your system meets the following requirements:

- Microsoft Windows 7 32-bit or 64-bit or Microsoft Windows Server 2008
- Microsoft .NET Framework 4.0
- Microsoft SQL Server 2008
- access to an administrator account

Components

The installation comprises of several components, some of them optional:

- XRouter Service
- XRouter Manager
- XRouter GUI (or XRouter Configuration manager GUI)
- DaemonNT console tool
- DaemonNT Configuration editor GUI
- Documentation package
- Examples

Program files

The most important program files are mentioned in the following list:

- `XRouter.ComponentHosting.dll` – XRouter Service
- `XRouter.Manager.dll` – XRouter Manager service
- `XRouter.Gui.exe` – XRouter Configuration manager GUI
- `DaemonNT.exe` – DaemonNT console tool – service installer and runner
- `DaemonNT.GUI.ConfigEditor.exe` – DaemonNT Configuration editor GUI
- `DaemonNT.xml` – configuration of the DaemonNT services
- `XRouter.Gui.exe.config` – configuration of the XRouter Configuration manager

Both `XRouter.ComponentHosting.dll` and `XRouter.Manager.dll` contain DaemonNT services and those assemblies are referenced from the configuration.

Each service needs the DaemonNT tool. XRouter Service can be run alone, just to perform the XML routing job. XRouter Manager service adds some monitoring and management features. XRouter Configuration manager GUI serves as an optional user interface to the XRouter Manager in order to configure the XRouter Service and see the history of its work. Similarly DaemonNT Configuration editor GUI provides an optional user-friendly way of configuring DaemonNT services.

The DLL files contain platform independent MSIL code, the EXE files are 32-bit.

Database

As a database server an instance of Microsoft SQL Server 2008 R2 is used. Both XRouter server and XRouter Manager share the same database by the name of XRouter into which they use the same login. For more information please see the Installation part of this document or the XRouter.Data chapter of the Programmer's Guide.

Log files

All the logs in the XRouter context are produced either by DaemonNT services or when installing them with the Windows service installer. Please refer to the DaemonNT documentation for more details. In brief DaemonNT services produce logs in the format `Logs/`*`YYYY_MM_DD_service-name`*`.[Trace.]log`, where the directory is relative to the `DaemonNT.exe` assembly location and currently is not configurable. There are two main log types: *event log* with some sparse information on service start up or shutdown or other important events and *trace log* in XML format with more detailed execution information and exceptions.

Logs can be viewed either in the raw text/XML form or via the XRouter Configuration manager GUI. The latter in addition provides filtering by date/time interval and log levels. You are free to use any log post-processing or archiving tools.

Documentation

The installation is provided with a set of documents located in the `Documentation/` directory:

- `README.txt` – basic project information
- `LICENSE.txt` – legal information regarding XRouter and its third-party components
- API documentation in the CHM format
- PDF originals of the printed documents

Installation

The order whether to install the database first or the program files first is not important. Just make sure you have the database prepared and accessible before you start the services or the XRouter Configration manager GUI.

Database

For most of the following operations it is best you use SQL Management studio which is a part of SQL Server, it is also assumed you are logged in the correct instance as an administrator. If you have more than one instance of Microsoft SQL Server on your server and the instance you intend to use for XRouter is not the default one, add the instance name into the connection string in DaemonNT configuration.

First, make sure your instance of the MS SQL Server has remote logins allowed:

1. right-click on the connected instance in the Management Studio
2. go to *Properties/Security*
3. choose *SQL Server and Windows Authentication Mode* and hit *OK*

Then you need to run the `CreateLogin` script which will create the `XRouter_AccessDB` login used by the application that has the correct password (that you may want to change, but make sure it corresponds with the one in the connection string) and SID which is important for easier database deployment. Make sure the newly created login is set as enabled:

1. in the left panel expand *Security/Logins*
2. right-click on the newly created login
3. go to *Properties/Status*
4. choose *Login*: *Enabled*, and hit *OK*

Then restore the provided `.bak` file and you are done, because thanks to the login having the corresponding SID as the login on the server the backup was made, all the security mappings are done automatically. To do the restore using Management Studio:

1. in the left panel right-click on *Databases*
2. choose *Restore database...*

3. in the dialog choose *From device* and click the "..." button
4. in the new dialog add the location of the `.bak` file, click *OK*
5. back in the previous dialog enter *XRouter* in the textbox labeled *To database:* and click *OK*

Further information about database usage may be found in the XRouter.Data chapter of the Programmer's Guide.

Installing program files

In order to install the program files and additional files into the operating system obtain and run the installer `xrouter-setup.exe`. Probably you will need administrator privileges to do the latter. There is not much surprising in the installer.

1. First you have to accept the license.
2. Select the location where XRouter should be installed. The default value depends on whether you are running 32-bit or 64-bit Windows, ie. `C:\Program Files\XRouter` or `C:\Program Files` **`(x86)`**`\XRouter`. Of course you can specify another location as well.
3. Specify the components which should be installed. The *Full installation* is suitable for most purposes, while the *compact installation* only contains files needed to run the XRouter Service and XRouter Manager, eg. on a separate remotely configured node. Moreover, you can select an arbitrary configuration and dependencies should be handled properly by the installer.

After the installation you can read the README file. The important commands are accessible from the Start Menu (named XRouter by default).

Installing Windows services

If you plan to run XRouter and/or XRouter manager service, you should install each one as a Windows service. It just creates a new identifier, associates it with a command and some metadata. It does not copy any files. The installer does not do this automatically since you might want some different service names or multiple service instances.

The easiest way is to click on the provided link in the Start Menu and run as Administrator: *Install XRouter as Windows Service* and *Install XRouter Manager as Windows Service*. Those links assume the default service names `xrouter` and `xroutermanager`. In case of multiple service instances or different names you have to use `DaemonNT.exe` directly:

- run `cmd.exe` (as Administrator)
- `cd C:\Program Files (x86)\XRouter` (or your main installation directory)
- `DaemonNT.exe install` *`xrouter_service_name`*
- `DaemonNT.exe install` *`xrouter_manager_service_name`*

Please refer to the DaemonNT documentation for more details.

Uninstalling

Make sure you stop all the services and other programs running from the installation location you are about to uninstall. Otherwise some program files and logs might still be locked and you would have to delete them by hand afterwards. Also if you are not just upgrading you should first uninstall the Windows services installed by DaemonNT (this is analogous to installing described above).

In order to run the uninstaller either run the *Uninstall XRouter* Start Menu link or directly the *unins000.exe* program. As with the installation you might need administrator privileges. Bear in mind that log file produced by the services will not be deleted.

Upgrading

Upgrading to a new version can be done by uninstalling the old version and installing the new one. Log files and configuration file are preserve, but still make sure you back-up the important files.

Configuration

Both DaemonNT services, XRouter Service and XRouter Manager, have to be configured before their usage. In this installation both share the same configuration file, `DaemonNT.xml`, located besides the `DaemonNT.exe` tool. One or more instances of each service can be configured there. The `DaemonNT.exe` tool enables to specify a different configuration file, but this is currently limited only to the debug mode. When running a DaemonNT service as a Windows service, the default configuration file is used. If you really needed a different file you would have to modify the command to run the Windows service by hand.

You can edit the configuration file by hand or use `DaemonNT.GUI.ConfigEditor.exe`, the DaemonNT Configuration editor GUI. Just be aware of that the `DaemonNT.xml` might be writable only with Administrator privileges. For details of the configuration file format and semantics please refer to the DaemonNT documentation.

XRouter Service

DaemonNT service configuration

The most important thing to configure here is the connection string for the database. It is located in the `connection-string` parameter of the `broker` section of the XRouter Service. Its format depends on the database used – currently this is just a MS SQL Server connection string. In case of the SQL Server authentication the database server IP or hostname, database name, user name and password has to be provided (see the example snippet).

You can create one or more instances of the XRouter Service here, each with a unique identifier. Just create multiple `<service>` elements, refer to the same assembly and class and provide a unique service name.

Example snippet:

```
<service name="xrouter"
type="XRouter.ComponentHosting.XRouterService,XRouter.ComponentHosting.dll">
    <settings>
      <section name="broker">
        <param name="connection-string">Server=192.168.1.1;Database=XRouter;User
Id=username;Password=secretpassword;</param>
      </section>
...
```

Application configuration

The main XRouter Service configuration is stored in the database (inside one big XML). It can be edited in the XRouter Configuration manager GUI, exchanged with the database and imported/exported from/to a plain XML file.

The application configuration is composed of several parts: configuration of each component (such as adapter instances for a gateway, the number of threads in processor, etc.), registration of adapter and action types, current message flow and a history of previous ones, XML resources.

Further application configuration of the behavior of the XRouter Service is up to the user and is described in detail in the XRouter – User's guide document.

XRouter Manager

This service provides monitoring, management and a watchdog for the XRouter Service. A single instance of the manager takes care of a single XRouter Service. It is possible to create serveral such pairs on one machine, you just need more databases. This service is entirely configured in the DaemonNT configuration file. The service is composed of several modules, each with its own configuration section.

Example:

```
<service type="XRouter.Manager.XRouterManagerService,XRouter.Manager.dll"
name="xroutermanager">
    <settings>
      <section name="email">
        <param name="smtpHost">192.168.1.1</param>
        <param name="smtpPort">25</param>
        <param name="from">xrouter-manager@example.com</param>
        <param name="to">xrouter-admin@example.com</param>
      </section>
      <section name="storages">
        <param name="connectionString">
          Server=192.168.1.1; Database=XRouter;
          User Id=username; Password=secretpassword;
        </param>
        <param name="logs">Logs</param>
      </section>
      <section name="watcher">
        <param name="autoStartEnabled">True</param>
      </section>
      <section name="console">
        <param name="uri">
          http://localhost:9090/XRouter.ConsoleService
        </param>
        <param name="metadataUri">
          http://localhost:9090/XRouter.ConsoleService.Metadata
        </param>
      </section>
      <section name="reporter">
        <param name="time">01:00:00</param>
      </section>
      <param name="managedServiceName">xrouter</param>
    </settings>
  </service>
```

Section `email`

An optional section for the SMTP e-mail sender used for sending watcher notifications and reports summarizing logs. In case the section is not configured no e-mails will be sent.

- `smtpHost` – host name or IP address of the SMTP server (required parameter)
- `smtpPort` – port on the host where the SMTP server is listening (optional); if none is specified the default is used: port 25 (according to RFC 821)
- `from` – e-mail address of the sender, ie. the notification service (required)
- `to` – e-mail address of the recipient, ie. the administrator (required)

At first make sure the SMTP server is accessible from the machine where XRouter runs (no firewalls and mail server rules are blocking the traffic) and that it accepts e-mails with the specified sender and recipient addresses.

Section `storages`

A required section with information needed to access some persistent resources, such as database or log files.

- `connectionString` – connection string for connecting to the XRouter database (required parameter); must be the same as for the managed XRouter Service instance
- `logs` – (relative or absolute) path to the directory containing DaemonNT log files produced by the managed XRouter Service (optional parameter); by default the `Logs` directory inside the directory of the XRouter Manager's assembly is used

Section `watcher`

A required section for configuring the watchdog which monitors and automatically restarts the managed service.

- `autoStartEnabled` – required boolean parameter (`true` or `false`)
    - `true` – in case the manager service is stopped for some time (if it fails, is stopped by mistake, etc.) the watcher will try to start the service again; if successful an e-mail notification will be sent
    - `false` – the watcher will not be active

Section `console`

A required section for configuring the management ConsoleServer (used by the GUI).

- `uri` – URI of the ConsoleServer web service (WCF) hosted inside the XRouter Manager (required parameter)
- `uriMetadata` – URI of the ConsoleServer web service (WCF) providing the metadata, ie. a description of the interface needed for creating a remote proxy

Section `reporter`

A required section for configuring the Reporter, which periodically (once a day at the specified time) scans recent log files and sends summary reports by e-mail.

- `time` – a TimeSpan specifying the time of the day when reports should be created and sent (required parameter)

Parameter `managedServiceName`

A system-wide identifier of the XRouter Service instance managed by this XRouter Manager instance (required parameter). This also corresponds to the `name` attribute of the `service` element of the XRouter Service configuration in the DaemonNT configuration file.

XRouter Configuration manager GUI

The GUI connects to a `ConsoleServer` web-service provided by a running instance of the XRouter Manager service. The URI of this web-service should be configured in the `XRouter.Gui.exe.config` file and corresponds to the URI specified in the XRouter

Manager configuration – section `console` – parameter `uri`.

Example:

```
<configuration>
  <appSettings>

<add key="consoleServerUri" value="http://localhost:9090/XRouter.ConsoleService/
ConsoleServer"/>
  </appSettings>
</configuration>
```

Running

There are several ways how to run the XRouter Service and its additional tools. However, for both the XRouter Service and XRouter Manager a MS SQL Server with the database need to be available. Please refer to its respective documentation. Also you should be familiar with how to work with the `DaemonNT.exe` tool – details are in the DaemonNT document.

In the most spartan case just the XRouter Service runs. This might be useful either for a developer testing modifications or new features but also to test that a particular application configuration works properly. In production usually more features are appropriate – the XRouter Service should be monitored and automatically restarted in case of a failure, logs should be scanned and summary reports made from them, etc. Those additional features are provided by the XRouter Manager service. In the third case the user or administrator might want to interact with the service, eg. via a GUI – at least edit and change the configuration.

XRouter Service

For testing purposes you can run XRouter Service without installing it as a Windows service. Just run `DaemonNT.exe debug` *`xrouter`* (where *`xrouter`* is the actual name of the service) – as Administrator if writing logs requires it. It is only suitable for debugging and testing, not for production as it is not managable by XRouter Manager.

For production purposes XRouter should be installed as a Windows service and run as such with `DaemonNT.exe start` *`xrouter`* or the *Start XRouter as Windows Service* link in the Start Menu.

XRouter Manager

Also in production it is better to use the XRouter Manager in addition to the XRouter Service. The Manager can be started while XRouter Service is running or independently. In case it is configured to automatically start a stopped XRouter Service you just might need to start the XRouter Manager alone. Run `DaemonNT.exe start` *`xroutermanager`* or the *Start XRouter Manager as Windows Service* link in the Start Menu.

XRouter Configuration manager GUI

In order to view, edit and change the application configuration of the XRouter Service or to view the logs you need to start the XRouter Configuration manager GUI. It depends on a running instance of the XRouter Manager. Run `XRouter.Gui.exe` or the *XRouter Configuration Manager* link from the Start menu. This program can be started as an ordinary user.

Running web-services as an ordinary user

In case of running a web-service as an ordinary user the system might want to explicitly allow the port and URI prefix. This applies to XRouter Manager and possibly some example web services. When running the same program as a Windows service there is no problem.

To allow a port, eg. *8011*, for a web-service for the *johndoe* user run the following command in `cmd.exe` with Administrator privileges:

```
netsh http add urlacl url=http://+:8011/ user=johndoe
```

Extending

In case you have extended some parts of XRouter you have a new assembly containing the new class and want to register it within XRouter. This is sometimes done inside the DaemonNT configuration file, sometimes in the aplication configuration stored in a database. In both cases new classes are referenced just like the class of a DaemonNT service itself (such as the XRouter Service) is referenced – by the CLR typ name and assembly name separated by a comma. Whitespace around or between is allowed.

The new assembly can be put anywhere and referenced via the absolute path, but it is not a bad idea to put the assembly in the directory where XRouter is installed or in a subdirectory, so that a simple relative path can be used.

For example if you have a new trace logger storage in the `DatabaseStorage` class within the `Foo.DatabaseStorage.dll` and you put it right into the XRouter program files directory, you can configure it like the following:

```
<config>
  <service name="xrouter" type="
      XRouter.ComponentHosting.XRouterService,
      XRouter.ComponentHosting.dll"
    <trace-logger>
      <storage type="DatabaseStorage,Foo.DatabaseStorage.dll"
        name="fooDatabaseStorage" />
    </trace-logger>
  </service>
...
```

Of course, to take effect of the new settings you have to restart the service using it.

Plugins with custom adapters and actions

Assemblies containing custom adapters should be places into the `AdapterPlugins` directory within the XRouter installation directory. Similarly custom actions are searched in the `ActionPlugins` directory. Only some essential built-in action plugins are searched for right in the installation.