Ascent360 Raptor Service - Technical Specifications, Communications Protocol, and Auto-Update

Ascent360 has many methods for collecting data securely from our clients. This includes batch SFTP, Webhooks, HTTPS POST, Native API and Web Services. In addition to these, Ascent360 has created a Windows Service, Raptor, that can collect data from any database system located on a windows network. This Article is specific to the Raptor Service.

In this article:

Raptor Installation Prerequisites

1. **Software Requirement**
    - .NET Framework 4.8.
    - If server operating system, Windows Server 2008 R2 or higher. If desktop operating system, Windows 7 or higher.
    - Can browse to https://webupload.ascent360.com
    - Can reach 20.184.240.134
    - Underlying database architecture is SQL Server. If not SQL Server, see sections regarding ODBC below.
2. **Hardware Requirement**
    - Processor: Minimum 1GHz
    - RAM: Minimum 2GB memory available for the application (Raptor Service)
    - Disk space: Minimum 5 GB

The .NET 4.8 requirement is especially important to verify **before the installation** if the machine that Ascent360 is installing the Windows Service on is production-facing. Upgrading the version of .NET requires a full restart of the machine which is not something we can do during the daytime (if the system is production / not a VM).

Installation Process

Installation ranges from 30 minutes to 1 hour. Ascent360 will host a shared meeting and walk through the install process with you and/or your IT Resource.

- Software: Ascent360 has an executable file that will install the Raptor Service. It is available in both 32 bit or 64 bit.
- Permissions: Typically, a SQL reader account needs to be created for Ascent360. This reader account can be limited to a specific set of tables and fields if desired.

Protocols, Processes, and Features

Connection Protocols: ODBC, Native SQL Server Driver, Native Oracle Driver.

Communication Protocol: The Raptor service will transmit data via SFTP or HTTPS. All communications occur using these protocols to ensure all messages and data are secured over an encrypted channel.

Communication Process:

The system runs 3 processes on a .NET timer. These processes include:

- Heartbeat: This process sends a short message to Ascent360 every 30 minutes to simply say that things are “OK”
- Get instructions: This gets new 'instructions' from Ascent360. 'Instructions' can be queries, timings or connection strings.
- Run Queries: This is the key process. This process will run each query that we need in sequential order.
    - Typically, our system will run 10 to 30 queries depending on the point of sale (POS) / eCommerce system.
    - Hold query result in memory. This system does not create any files or store any data on the disk to ensure security.
    - Convert data to XML and stream to Ascent360 via HTTPS (256-SSL or SFTP as noted).

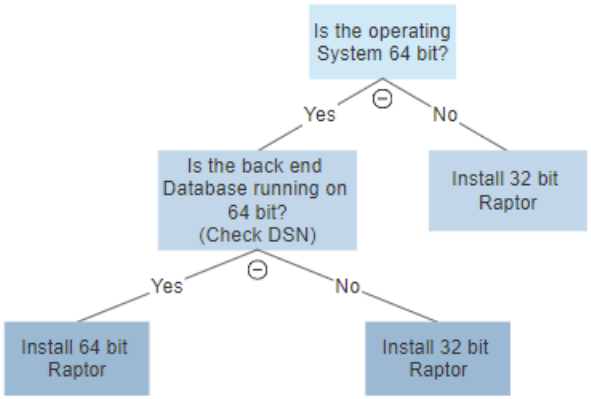
Communication Ports: To transfer data via HTTPS, we will need to have Port 443 open.

Location of the install:

The Ascent360 Raptor Service can be installed on any windows server that has database access, such as the Siriusware or RTP database. Our service will use the SQL reader account to query the database and deliver this data to the Ascent360. In small implementations, the Raptor Service may be installed on the same server as the database. In larger implementations, the service may be installed on another server or even a dedicated server.

32-bit vs 46-bit install:

As the Raptor service is available in both 32 bit and 64 bit version, please refer to the following decision tree to inform which version to use.



In some cases, it is possible that we may be using ODBC drivers if the database is not SQL Server. In that case, ODBC drivers version also needs to be checked to decide which version of Raptor windows service needs to be used.

ODBC 32-bit or 64-bit?

Timing of Data Transfer:

- Data Transfer: Typically, our clients will choose to send data to Ascent360 once per day. The timing of this can be at the clients request, but is usually done at night. Data can be transferred as often as every 15 minutes.
- Heartbeat: As a matter of quality control and monitoring, our service will send an HTTPS POST “Heartbeat” to Ascent360 every 10 minutes to check-in but it's configurable and can be changed to as low as 1 min.

Auto-Update:

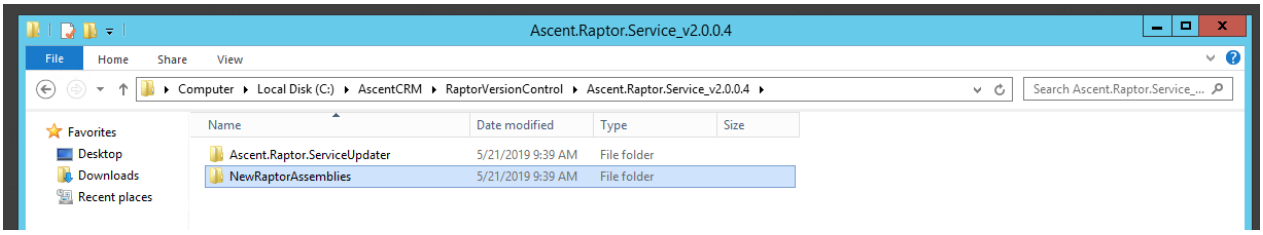
The Ascent360 Raptor Service has the ability to auto-update. Ideally, the client should only have to install the Raptor Service one time. Moving forward, based on a specific criteria, the following tasks will be performed - request for updated version, create backup directory, shut the Raptor Service down, replace old version with new, restart the Raptor Service, and perform a fail-back if any issues are encountered. Each performed task will be covered in more detail below.

- Specific Criteria: Ascent360's databases host two configuration tables - Raptor.Config & Raptor.VersionControl. Within these tables, the user can find various fields specific to the client. The auto-update feature is triggered based on conditions of the following fields within Raptor.Config - AutomaticUpdate, UpdateToVersion, and UpdateVersionOnDate. These conditions are checked at the time the Raptor Service starts, and during heartbeat check-ins. Once an update is deemed ready, the Raptor Service will also verify that no processes are currently being executed before initiating the auto-update feature.

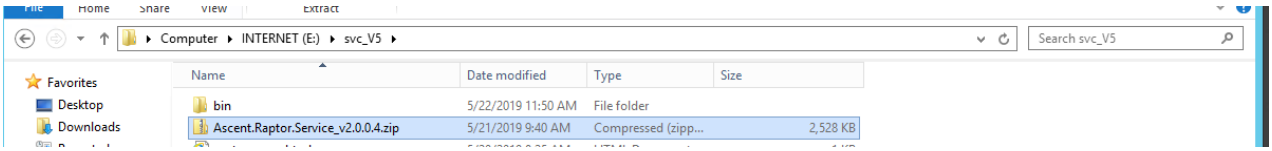
- Request for Updated Version: Within the Raptor.VersionControl table, the user will find a field - RaptorLatestVersionPath. This path will contain the specific .zip for a particular version. Configuring the tables in this manner gives Ascent360 and the client granular control over when to update, and which version to update to. The Raptor Service, once ready, will make a HTTPs request for the .zip file at this specific location. It

is important to mention, this path is on the same server as the Raptor.Service.Web API. Within the .zip file, two folders will exist, one with the Raptor ServiceUpdater tool and the other with the new version of the Raptor Service.

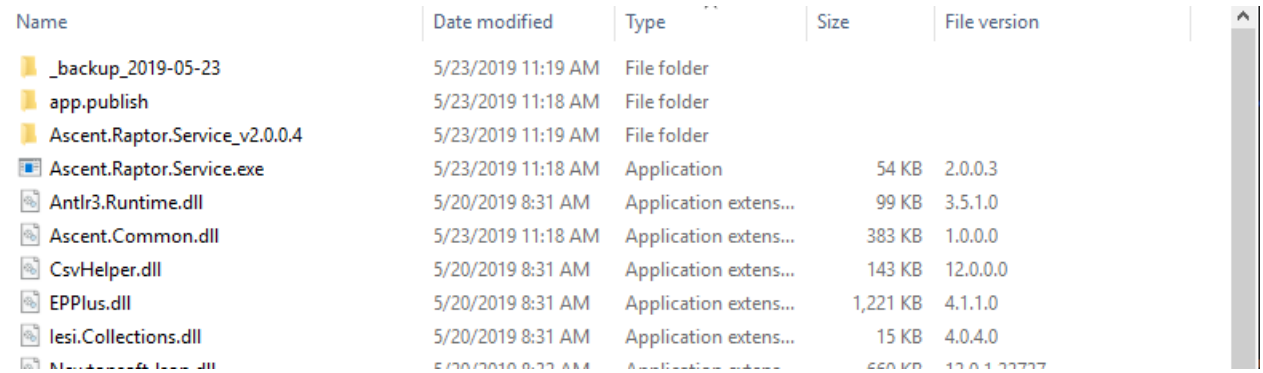
Note: Within the .zip file



Note: The .zip file



- Create Backup Directory: Once the Raptor Service, on the client's machine, has received this package. It will perform three tasks - extract the received package into its own directory (will be same name as .zip), create a backup directory, and copy the existing version into this newly created backup directory. Below is how the Raptor Service directory, on the client side, should appear after this process.



- Shut down, replace, restart: At this point, the Raptor Service will start an application included in the update package called Raptor ServiceUpdater, and shut itself (the Raptor Service) down. Everything will now occur within Raptor ServiceUpdater and will not be seen by the client. The Raptor ServiceUpdater will first wait 30 seconds, this will give the Raptor Service enough time to fully stop, and then replace the old assemblies with the new. Once verified, the Raptor ServiceUpdater will restart the newly updated Raptor Service and shut itself down.

- Fail-back: The Raptor ServiceUpdater contains functionality which will retry certain actions at a specified interval to ensure reliability and provide self corrective capabilities. In the case that the Raptor service could never be stopped - no action will be taken, and the client will have to manually restart the Raptor Service if the status is anything other than 'Running'. In the case that the new assemblies were updated, but the Raptor Service could not be restarted - the backed up version will be put back in its original directory. Email capabilities have been added to notify the Application Development team at Ascent360 of such issues. A manual restart, at this point, will be necessary.