Encrypting Destiny® data files

Protecting data on the Destiny application server for v9.0 and higher

Overview

This content describes how a system administrator can further protect application data on the Destiny application server by encrypting external Destiny data files.

Certain Destiny data files are stored and served from standard drive folders on the Destiny application server. This data includes text documents such as patron import files and report output as well as image files such as patron pictures. These files are external to the SQL database and as such are to be protected from unauthorized access with normal Windows access controls—logon authentication and file permissions.

External files can still be accessible to physical access such as theft of the server or hard drive. Hard drive protection by normal logon access control can be bypassed by booting a separate operating system or plugging the hard drive into another computer.

Protecting with Encryption

Some customers may wish to protect Destiny files against such physical access. You can do this with file encryption. With file encryption, files are stored as unintelligible characters and are therefore protected even when an attacker has full physical access to the hard drives. Even a remote user session cannot read the files if that account has not been granted encryption rights to them.

Using the Encrypted File System

The encryption mechanism recommended for use with Destiny is the Encrypted File System (EFS) technology built into Windows Server®. EFS allows the system administrator to designate that certain files should be encrypted whenever they are saved to disk. Because EFS is integrated with NTFS, the encryption process occurs automatically and is transparent to authorized user accounts. Furthermore, there are no third-party add-ons to install and manage.

According to Microsoft, "Only authorized users and designated data recovery agents can decrypt encrypted files. Other system accounts that have permissions for a file—even the Take Ownership permission—cannot open the file without authorization. Even the administrator account cannot open the file if that account is not designated as a data recovery agent. If an unauthorized user tries to open an encrypted file, access will be denied."

EFS encryption strength

EFS uses industry-standard public-private key technology to provide strong encryption. Windows Server uses the Advanced Encryption Standard (AES) algorithm by default, which uses a 256-bit key for encryption and decryption. The encrypting/decrypting process is performed in kernel mode, eliminating the risk of keys being left in an external paging file.

Encrypting Destiny data

EFS encryption can be set at the folder level so that all files created in that folder are automatically encrypted. In Destiny, the FSC-Destiny folder is the parent folder under which non-SQL data files are stored. Assigning EFS encryption to this folder and below will encrypt external Destiny files.

You can use an existing user account on the server to encrypt the Destiny folder, or create a new one specifically for this purpose. The steps below assume you will be creating a new account.

IMPORTANT Be sure to stop the Destiny service before performing these steps.

Create a new user account on the server for the Destiny service

1. Select **Start > Control Panel > Administrative Tools > Computer Management**. The Microsoft Management Console opens.
2. On the left-hand side, expand the Local Users and Groups tree and select the Users folder.
3. In the right-hand window, right-click and select New User.
4. Enter a user name and password and set the options for the new account.

   Make sure you select the following options:

   - Password never expires
   - User cannot change password

   IMPORTANT Do **not** select these options: *User must change password at next logon* and *Account is disabled*.

5. Click **Create**.
6. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.
7. On the left-hand side, expand the Local Policies tree and select **User Rights Assignment**.
8. On the right-hand side, double-click **Log on as a service**.
9. Click **Add User** or **Group**.
10. Select the Destiny account.
11. Click **OK**.

Modify the Destiny service to run under the new account

1. Unregister the Destiny Service, as follows:

   - Open a command prompt and navigate to the `\FSC-Destiny\jboss\bin` directory.
   - Run the following command: `destiny unregister`

2. In the `\FSC-Destiny\fsc\bin` directory, create a plain text file named password.conf and add the following lines:

   ```
   wrapper.ntservice.account=<domain name>\<user name>

   wrapper.ntservice.password=<password>
   ```

   NOTE If the server is not part of a domain, use the machine name instead of a domain name.

3. Log in to Windows Server using the Destiny account.
4. Register the Destiny Service, as follows:

   - Open a command prompt and navigate to the \FSC-Destiny\jboss\bin directory.
   - Run the following command: `destiny register`
   - Close the command prompt.

Update the folder properties on `\FSC-Destiny`

1. Log in to the Windows server using an Administrator account.
2. Open Windows Explorer and navigate to the `\FSC-Destiny` folder.
3. Right-click on the `\FSC-Destiny` folder and select **Properties**.
4. On the *Security* tab, add the Destiny user account and grant it Full Control.

5. On the *General* tab, click **Advanced**.
6. Select **Encrypt contents to secure data**.
7. Click **OK**.
8. On the *General* tab, click **Apply**. The *Confirm Attribute Changes* box opens.
9. Select **Apply changes to this folder, subfolders and files**
10. Click OK.
11. Restart the Destiny service.

At this point, you should have no unencrypted data in your Destiny installation. Be sure to check for any Destiny-related data (such as patron upload files) that may have been stored elsewhere on the server and move it to the FSC-Destiny directory if necessary. If you prefer to keep these files outside the FSC-Destiny folder, you can encrypt the folder where they are stored using the same Destiny account.

When running command-line utilities (such as patron or class uploads) from an encrypted folder, be sure to first log in to the Windows server using the Destiny account. When running command-line utilities from an encrypted folder as a scheduled task, be sure to configure the scheduled task to run as the Destiny account.

Backup notes

This process only encrypts the contents of the folder where Destiny is installed. If you copy the contents of the Destiny directory to another location, the files may not be encrypted in the new location. This is important to keep in mind when making backups of Destiny. You may want to encrypt your backup location as well, using the same steps as above. In case of server malfunction, you may also want to export the certificate that you used to encrypt the folder. For more information on this process, contact Microsoft technical support.

Technical Support

Destiny

Email techsupport@fsc.follett.com

Phone Support 800.323.3397

Follett Shelf

Email follettshelfsupport@follett.com

Phone Support 877.873.2764

Customer Service

Email customerservice@fsc.follett.com

Phone 800.323.3397 or 815.344.8700