Data Conversion in Windows: A Comprehensive Guide

Data conversion is a critical aspect of working with computer systems, as it involves transforming data from one format to another. In the Windows environment, data conversion plays a crucial role in various scenarios, such as migrating data between different applications, importing and exporting data, and ensuring compatibility between different systems.

One of the primary challenges in data conversion is dealing with different file formats. Windows provides several built-in tools and utilities that can be used to convert data between formats. For example, the Command Prompt (CMD) and PowerShell offer powerful command-line options to manipulate and convert data.

Examples:

1. Converting CSV to Excel:

   - Open the Command Prompt or PowerShell.
   - Use the following command to convert a CSV file to an Excel file:

     ```
     powershell -c "Import-Csv C:\input.csv | Export-Excel C:\output.xlsx"
     ```

   - Replace "C:\input.csv" with the path to your CSV file and "C:\output.xlsx" with the desired output Excel file path.

2. Converting XML to JSON:

   - Open the Command Prompt or PowerShell.
   - Use the following command to convert an XML file to a JSON file:

     ```
     powershell -c "Get-Content C:\input.xml | ConvertTo-Json | Out-File C:\output.json"
     ```

   - Replace "C:\input.xml" with the path to your XML file and "C:\output.json" with the desired output JSON file path.

3. Converting Text to UTF-8:

   - Open the Command Prompt or PowerShell.
   - Use the following command to convert a text file to UTF-8 encoding:

     ```
     powershell -c "$content = Get-Content C:\input.txt; [System.IO.
     ```

```
File]::WriteAllLines('C:\output.txt', $content, [System.Text.Encoding]::UTF8)"
```

- Replace "C:\input.txt" with the path to your text file and "C:\output.txt" with the desired output file path.