

# Natural Web I/O Interface Client

Der Natural Web I/O Interface Client wird in einer Remote-Entwicklungsumgebung benutzt, um Nicht-GUI-Informationen anzuzeigen, die Unicode-Zeichen enthalten. Dies ist anders als in der lokalen Umgebung, in der die Ausgabe in einem Ausgabefenster angezeigt wird.

Dieses Kapitel umfasst die folgenden Themen:

- Über den Natural Web I/O Interface Client
- Anforderungen für den Natural Development Server
- Mit dem Natural Web I/O Interface Client arbeiten
- Farb- und Feldattribute ändern

Siehe auch *Differences between Natural Web I/O Interface Client and Terminal Emulation* in der *Natural Web I/O Interface*-Dokumentation.

## **Wichtig:**

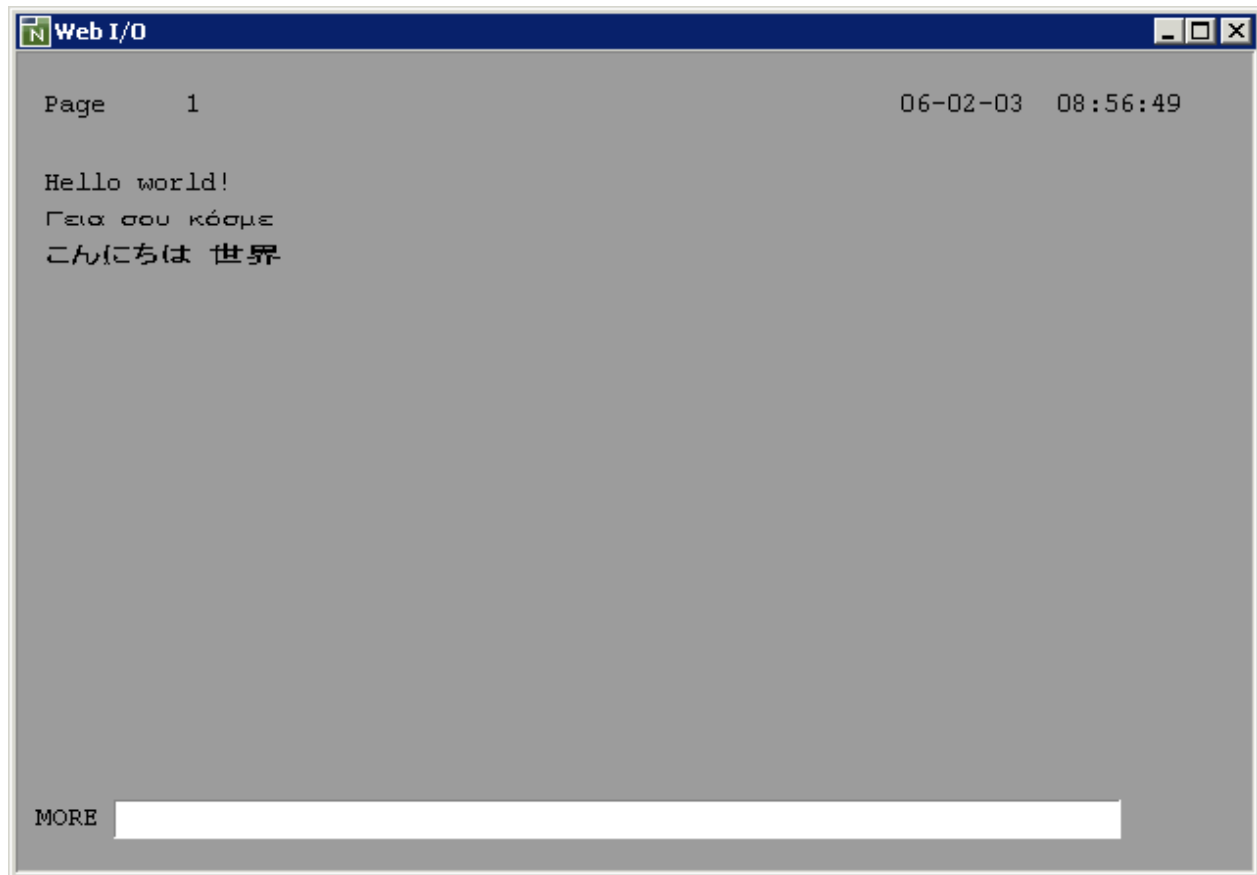
Um den Natural Web I/O Interface Client unter Microsoft Windows auszuführen, muss die Personal Firewall deaktiviert werden. Weitere Informationen finden Sie unter *Configuring the Microsoft Windows Personal Firewall to Run Natural* in der *Operations*-Dokumentation.

---

## Über den Natural Web I/O Interface Client

Wenn bestimmte Parameter auf dem Natural Development Server gesetzt wurden (siehe unten), wird der Natural Web I/O Interface Client anstatt des Terminal-Emulationsfensters benutzt (das Terminal-Emulationsfensters ist nicht Unicode-fähig).

Das folgende Beispiel zeigt einen Natural Web I/O Interface Client, der die Ausgabe eines Programms enthält, das in einer Remote-Umgebung ausgeführt wurde.



Der Natural Web I/O Interface Client ermöglicht es Ihnen, eine vollständige Anwendung zu testen, einschließlich aller ihrer Zeichen-Ausgaben, ohne dass Sie Natural Studio verlassen müssen. Eine Bildschirm-Ausgabe eines ausgeführten Programms bewirkt, dass der Natural Web I/O Interface Client automatisch geöffnet und wieder geschlossen wird, wenn das Programm beendet wird (logisch gesehen erreicht die Entwicklungs-Server-Session die NEXT-Ebene).

Utilities, für die Natural Studio keine graphische Benutzeroberfläche in der Remote-Umgebung anbietet, werden durch den Natural Web I/O Interface Client auch bereitgestellt. Die Eingabe des betreffenden Systemkommandos in der Kommandozeile von Natural Studio bewirkt, dass der Natural Web I/O Interface Client automatisch geöffnet wird.

Wenn Sie den Natural Web I/O Interface Client manuell schließen, entspricht dies dem Drücken von ESC in der Anwendung.

## Anforderungen für den Natural Development Server

Damit der Natural Web I/O Interface Client aufgerufen werden kann, muss der Natural Development Server (NDV) wie folgt konfiguriert werden:

- **UNIX und OpenVMS**

Wenn Sie den Natural Web I/O Interface Client in einer Remote-UNIX- oder OpenVMS-Umgebung benutzen möchten, muss der Profilparameter WEBIO auf dem NDV-Server auf "ON" gesetzt werden. Siehe *Configuration Utility* in der Natural für UNIX- oder Natural für OpenVMS-Dokumentation.

- **Großrechner**

Wenn Sie den Natural Web I/O Interface Client in einer Remote-Großrechner-Umgebung benutzen möchten, muss der NDV-Konfigurationsparameter `TERMINAL_EMULATION` auf dem NDV-Server auf "WEBIO" gesetzt werden. Siehe *NDV Configuration Parameters* in der Natural Development Server-Dokumentation. Außerdem muss das Web I/O Terminal-Konvertiermodul `NATWEB` mit dem Natural-Nukleus verknüpft werden. Der Natural-Profilparameter `TMODEL` kann benutzt werden, um die Bildschirmgröße des Benutzers festzulegen.

- **Windows**

In einer Remote-Windows-Umgebung wird immer der Natural Web I/O Interface Client benutzt, ungeachtet der Einstellung des Profilparameters `WEBIO`.

## Mit dem Natural Web I/O Interface Client arbeiten

Wenn die Natural-Ausgabe in dem Natural Web I/O Interface Client gezeigt wird, erscheinen alle definierten PF-Tasten als Schaltflächen in dem Fenster (anstatt der Tasteneingabeaufforderungszeilen, die normalerweise in der ursprünglichen UNIX-, OpenVMS- oder Großrechner-Umgebung erscheinen). Sie können entweder diese Schaltflächen benutzen, oder Sie können die betreffenden Tasten auf Ihrer Tastatur drücken (zum Beispiel F5 für PF5).

### **Anmerkung:**

Eine Aktualisierungsfunktion wird nicht ausgeführt, wenn Sie F5 drücken.

Sie können Ihre Eingaben wie üblich vornehmen. Sie können auch die übliche Windows-Funktionalität Kopieren-und-Einfügen (Copy-and-Paste) einsetzen.

## Farb- und Feldattribute ändern

Im Natural Web I/O Interface Client werden die Farbattribute der Natural-Felder (Vordergrundfarben), die Hintergrundfarbe und die Darstellung der Natural-Feldattribute über eine XSLT-Datei gesteuert. Der Name dieser XSLT-Datei ist *natunistyle.xml*. Nach der Installation von Natural für Windows finden Sie diese Datei im Ordner *bin*.

Die folgenden Themen werden nachfolgend behandelt:

- Vordergrundfarben definieren
- Hintergrundfarbe definieren
- Attribut für inversen Text definieren
- Attribute für unterstrichenen und blinkenden Text definieren
- Attribut für kursiven Text definieren
- Attribut für intensivierten Text definieren

## Vordergrundfarben definieren

Die Vordergrundfarben für die Schriftart werden im folgenden Abschnitt der XSLT-Datei definiert:

```
<!-- Map foreground colors -->
<xsl:variable name="forecol">
  <xsl:choose>
    <xsl:when test="@forecol='Red'">red</xsl:when>
    <xsl:when test="@forecol='Green'">LimeGreen</xsl:when>
    <xsl:when test="@forecol='Blue'">DodgerBlue</xsl:when>
    <xsl:when test="@forecol='White'">white</xsl:when>
    <xsl:when test="@forecol='Pink'">fuchsia</xsl:when>
    <xsl:when test="@forecol='Turquoise'">aqua</xsl:when>
    <xsl:when test="@forecol='Yellow'">yellow</xsl:when>
    <xsl:when test="@forecol='Black'">white</xsl:when>
    <xsl:when test="@forecol='Transparent'">
      <xsl:choose>
        <xsl:when test="@intensified='True'">aqua</xsl:when>
        <xsl:otherwise><xsl:value-of select="$defforecolour"/></xsl:otherwise>
      </xsl:choose>
    </xsl:when>
    <xsl:otherwise><xsl:value-of select="$defforecolour"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

Beispiel: das Farbattribut Red des Natural-Bildschirms wird in die CSS-Farbe "red" umgesetzt:

```
<xsl:when test="@forecol='Red'">red</xsl:when>
```

Es ist auch möglich, RGB-Werte für die Farbattribute zu definieren. Beispiel:

```
<xsl:when test="@forecol='Red'">#FF0000</xsl:when>
```

Beim Farbattribut Transparent wird geprüft, ob das Attribut intensified gesetzt ist. Wenn es auf "True" gesetzt ist, wird die Farbe "aqua" benutzt. Wenn es nicht auf "True" gesetzt ist, wird die globale Variable \$defforecolour benutzt, die am Anfang der XSLT-Datei definiert ist.

Weitere Informationen zu den CSS-Farben können Sie im Internet finden, zum Beispiel unter [http://www.w3schools.com/css/css\\_colors.asp](http://www.w3schools.com/css/css_colors.asp) (in englischer Sprache).

## Hintergrundfarbe definieren

Die Hintergrundfarbe wird mit der globalen Variable defbackcolour definiert, die am Anfang der XSLT-Datei zu finden ist.

```
<!-- Background color -->
<xsl:variable name="defbackcolour" select="'black'"/>
```

## Attribut für inversen Text definieren

Das Feldattribut für die farbliche Umkehrung von Hintergrund und Feldwert wird im folgenden Abschnitt der XSLT-Datei definiert (er befindet sich direkt hinter dem Abschnitt, in dem die Vordergrundfarben definiert werden):

```

<!-- START SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->
<!-- Set foreground colors -->
<xsl:variable name="ForeCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">color:<xsl:value-of select="$defbackcolour"/></xsl:when>
    <xsl:otherwise>color:<xsl:value-of select="$forecol"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>

<!-- Set background colors -->
<xsl:variable name="BackCol">
  <xsl:choose>
    <xsl:when test="@reverse='True'">background-color:<xsl:value-of select="$forecol"/></xsl:when>
    <xsl:otherwise>background-color:<xsl:value-of select="$defbackcolour"/></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<!-- END SET FOREGROUND AND BACKGROUND COLORS FOR REVERSE ATTRIBUTE -->

```

Wenn das Attribut `reverse` auf "True" gesetzt ist, wird die Vordergrundfarbe auf den Wert von `$defbackcolour` gesetzt, und die Hintergrundfarbe erhält den Wert der Vordergrundfarbe.

Wenn das Attribut `reverse` nicht auf "True" gesetzt ist, erhält die Vordergrundfarbe den Wert der aktuell definierten Vordergrundfarbe, und die Hintergrundfarbe erhält den Wert von `$defbackcolour`.

## Attribute für unterstrichenen und blinkenden Text definieren

Die Feldattribute für unterstrichenen und blinkenden Text werden im folgenden Abschnitt der XSLT-Datei definiert:

```

<!-- Set underline and blinking attributes -->
<xsl:variable name="TextDekoStyle">
  <xsl:choose>
    <xsl:when test="@underline='True'">text-decoration:underline;</xsl:when>
    <xsl:when test="@blinking='True'">text-decoration:blink;</xsl:when>
    <xsl:otherwise>text-decoration:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>

```

Da der Natural Web I/O Interface Client den Internet Explorer benutzt, der blinkenden Text nicht unterstützt, kann blinkender Text nicht im Browser-Fenster angezeigt werden.

## Attribut für kursiven Text definieren

Das Feldattribut für kursiven Text wird im folgenden Abschnitt der XSLT-Datei definiert:

```

<!-- Set cursive attribute -->
<xsl:variable name="FontStyle">
  <xsl:choose>
    <xsl:when test="@italic='True'">font-style:italic;</xsl:when>
    <xsl:otherwise>font-style:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>

```

## Attribut für intensivierten Text definieren

Das Feldattribut für intensivierten Text wird im folgenden Abschnitt der XSLT-Datei definiert:

```
<!-- Set intensified attribute - draw text as bold -->
<xsl:variable name="FontWeight">
  <xsl:choose>
    <xsl:when test="@intensified='True'">font-weight:bolder</xsl:when>
    <xsl:otherwise>font-weight:normal;</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```