

Appendix H: SDFX File Format Definition

- Introduction
 - Mapset File Structure
 - Defining Maps
 - Defining Identifiers
 - Defining Fields
 - Defining Map Steps
 - Example of an SDFX File
-

Introduction

SDFX (Screen Definition Format) is a proprietary generic format used for describing a host screen. In addition to standard formats like BMS, MFS and NATURAL maps, ApplinX supports this generic format. If your host screen maps are not supported in ApplinX you can convert them to the SDFX format and then import them into your ApplinX repository.

SDFX is based on the XML structure. Each SDFX file is composed of one mapset, and must have the “.SDFX” extension. A mapset contains one or more smaller units called maps. Each map currently corresponds to one host screen. A map is made up of identifiers, fields, and map steps. Mapsets, maps, and fields are identified by name. ApplinX uses the map name to create a screen entity with the same name.

Mapset File Structure

The following details the mapset file structure, where all the map and field details are placed within the Map tag and are detailed in Defining Maps and Defining Fields.

```
<Mapset>
  <Name>mapsetname</Name>
  <Version>1</Version>
  <Type>SIMPLE</Type>
  <Map>
    ...
  <Map>
  <Map>
    ...
  <Map>
</Mapset>
```

Mapset Parameters

```
<Name>mapsetname</Name>
```

The name of the mapset (mandatory).

<Version>1</Version>

This line contains the SDFX format version (currently, always 1).

<Type>SIMPLE</Type>

This line contains the type of SDFX used. Currently the only type available is "SIMPLE".

<Map> ... <Map>

Refer to Defining Maps. A number of maps can be added to the same file.

Defining Maps

Each map represents a single screen and contains the screen properties such as the screen name, screen position, screen width and height etc.

```
<Map>
  <Name>MapName</Name>
  <PosX>1</PosX>
  <PosY>1</PosY>
  <Width>80</Width>
  <Height>24</Height>
  <IsPopup>>false</IsPopup>
  <Field class="SimpleField">
    ...
  </Field>
  <Field class="ListField">
    ...
  </Field>
  <LoopField>
    ...
  </LoopField>
</Map>
```

Map Parameters

<Name>MapName</Name>

The name of the map (mandatory).

<PosX>1</PosX>

The map's horizontal position (the X coordinate of the rectangle's top-left character, can be from 1 to the width of the screen). When not specified, the default is 1.

<PosY>1</PosY>

The map's vertical position (the Y coordinate of the rectangle's top-left character, can be from 1 to the height of the screen). When not specified, the default is 1.

<Width>80</Width>

The map's width (optional, default is 80).

```
<Height>24</Height>
```

The map's height (optional, default is 24).

```
<IsPopup>false</IsPopup>
```

True indicates that this map is used as a pop-up on the host (optional, default is false).

```
<Field class="SimpleField"> ... </Field>
```

Refer to Defining Fields. A number of fields can be added to the same map.

Defining Identifiers

Use SDFX identifiers to create ApplinX identifiers (refer to Defining Identifiers).

Parameters to use for a Text identifier

The identifier is recognized if its text matches (or does not match) the host screen text in the defined position.

Text

The string to try and match.

IsMatch

Boolean (true \ false), default value is true. When set to false, then the identifier is recognized only if the host screen's text \ attribute in that position does not match the identifier's text \ attribute.

IsCaseSensitive

Boolean (true \ false). When set to true, then the text will be case sensitive.

ScreenArea

Represents the screen area within in which to search for the identifier. Possible values:

Anywhere

Any position on the screen is a valid position.

For example

```
<ScreenArea class="Anywhere"/>
```

Position

Defines a position on the screen. StartPos determines the first position within the screen.

For example:

```
<ScreenArea class="Position">
  <StartPos>
    <PosX>40</PosX>
    <PosY>23</PosY>
  </StartPos>
</ScreenArea>
```

PositionLength

Defines the first position within the screen and the length of the area.

For example:

```
<ScreenArea class="PositionLength">
  <StartPos>
    <PosX>40</PosX>
    <PosY>23</PosY>
  </StartPos>
  <Length>15</Length>
</ScreenArea>
```

Rectangle

Defines a rectangle on the screen. StartPos determines the top left position and EndPos determines the bottom right position. Together these create a rectangle.

For example:

```
<ScreenArea class="Rectangle">
  <StartPos>
    <PosX>10</PosX>
    <PosY>1</PosY>
  </StartPos>
  <EndPos>
    <PosX>30</PosX>
    <PosY>24</PosY>
  </EndPos>
</ScreenArea>
```

The following example is sample code used for a text identifier, using the type "Anywhere".

```
<Identifier class="Text">
  <Text>TextToMatch</Text>
  <IsMatch>true</IsMatch>
  <IsCaseSensitive>true</IsCaseSensitive>
  <ScreenArea class="Anywhere"/>
</Identifier>
```

Parameters to use for an Attribute identifier

The identifier is recognized if its attribute matches the attribute that applies for the host screen in the defined position. The position does not necessarily has to be the position of the attribute character.

Attribute

The attribute to search for. Possible values: PROTECTED, HIDDEN, INTENSIFIED, and REVERSED_VIDEO.

ScreenArea: Position

Defines a position on the screen where the attribute should be checked.

For example:

```
<ScreenArea class="Position">
  <StartPos>
    <PosX>40</PosX>
    <PosY>23</PosY>
  </StartPos>
</ScreenArea>
```

The following example is sample code used for an attribute identifier (defines an "Unprotected" attribute).

```
<Identifier class="Attribute">
  <Attribute>PROTECTED</Attribute>
  <IsMatch>>false</IsMatch>
  <ScreenArea class="Position">
    <StartPos>
      <PosX>40</PosX>
      <PosY>23</PosY>
    </StartPos>
  </ScreenArea>
</Identifier>
```

Defining Fields

- Definitions of a Simple Field: Simple fields can be protected, unprotected or static fields.
- Definitions of a List Field: A List Field is a field which contains a repetitive pattern of simple fields (a "table" of similar cells).
- Definition of a Loop Field: A loop field describes a repetitive element of the host screen, which is not a list.

Definitions of a Simple Field

A simple field may be a protected, unprotected or status field. A field for which no name is given, will be considered by ApplinX to be an identifier. A simple field can also be used as a static identifier.

```
<Field class="SimpleField">
  <Type>SIMPLE</Type>
  <Name>DFHM001</Name>
  <PosX>2</PosX>
  <PosY>1</PosY>
  <Length>4</Length>
  <Text>string</Text>
  <Attribute>PROTECTED</Attribute>
  <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
  <Bright>NORM</Bright>
  <ForegroundColor>WHITE</ForegroundColor>
  <BackgroundColor>DEFAULT</BackgroundColor>
  <HighlightBlink>>false</HighlightBlink>
  <HighlightReverse>>false</HighlightReverse>
  <HighlightUnderline>>false</HighlightUnderline>
</Field>
```

Note:

A field which does not have a name tag but has a text tag, will be used as a screen identifier.

Simple Field Parameters

<Field class="SimpleField">

Indicates that this field is a simple field (a field's class can be SimpleField or ListField).

<Type>SIMPLE</Type>

Indicates that it's a simple field (optional, since SIMPLE is the default field type).

<Name>DFHM001</Name>

The name of the field (optional). A field for which no name is given will be considered by Applinx to be an identifier (a static area on the screen).

<PosX>2</PosX>

The field's horizontal position (the X coordinate of the position of the attribute character, can be from 1 to the width of the screen), relative to the original map.

<PosY>1</PosY>

The field's vertical position (the Y coordinate of the position of the attribute character, can be from 1 to the height of the screen), relative to the original map.

<Length>4</Length>

The field's length (not including the attribute character).

<Text>string</Text>

The initial value of the field (optional). This should match the field's length and will be truncated or space-padded as needed.

<Attribute>PROTECTED</Attribute>

The field's attribute. Possible values are: "UNPROTECTED" or "PROTECTED". The default value is "PROTECTED".

<DataType>DATA_TYPE_ALPHANUMERIC</DataType>

Determines whether the field can receive numeric or alpha numeric data. Possible values: "DATA_TYPE_ALPHANUMERIC" or "DATA_TYPE_NUMERIC". The default value is "DATA_TYPE_ALPHANUMERIC".

<Bright>NORM</Bright>

The field's attribute (brightness) (optional). Possible values: "BRT", "NORM" or "DRK". The default value is "NORM".

<ForegroundColor>WHITE</ForegroundColor>

Indicates the foreground color of the field (optional). The default color is white. The color must be written in capital letters. Possible values include: WHITE, BLACK, AQUA, BLUE, BROWN, GRAY, GREEN, LIGHT_AQUA, LIGHT_BLUE, LIGHT_GREEN, LIGHT_PURPLE, LIGHT_RED, LIGHT_WHITE, PURPLE, RED, YELLOW, NONE, DEFAULT.

<BackgroundColor>DEFAULT</BackgroundColor>

Indicates the foreground color of the field (optional). The default color is black. The color must be written in capital letters. Possible values include: WHITE, BLACK, AQUA, BLUE, BROWN_YELLOW, GREEN, PURPLE, RED, NONE, DEFAULT.

<HighlightBlink>>false</HighlightBlink>

BLINK highlighting (optional, default is false).

<HighlightReverse>>false</HighlightReverse>

REVERSE highlighting (optional, default is false).

<HighlightUnderline>>false</HighlightUnderline>

UNDERLINE highlighting (optional, default is false).

Definitions of a List Field

A List Field is a field which contains a repetitive pattern of simple fields. Since this repetition can expand vertically to the right (not only downward), a list can be seen as a “table” of similar cells (Applinx multiple fields \ tables). Each cell in the list is called a record and can contain several fields. You are required to define the first record (cell) and it must consist of Simple fields only.

```
<Field class="ListField">
  <Type>LIST</Type>
  <Name>DFHM002</Name>
  <PosX>1</PosX>
  <PosY>1</PosY>
  <Direction>HORIZONTAL_FIRST</Direction>
  <RecordWidth>1</RecordWidth>
  <RecordHeight>1</RecordHeight>
  <RecordsNum>4</RecordsNum>
  <Field>
    ...
  </Field>
  <Field>
    ...
  </Field>
</Field>
```

List Field Parameters

<Field class="ListField">

Indicates that this field is a list field (a field’s class can be SimpleField or ListField).

<Type>LIST</Type>

Indicates that it's a list field.

<Name>DFHM002</Name>

The name of the list (optional).

<PosX>1</PosX>

The list's horizontal position (the X coordinate of its top-left corner, can be from 1 to the width of the screen). Note that the corner of the list's area isn't necessarily the corner of the top-left cell (unless the cell's position is 1,1).

<PosY>1</PosY>

The list's vertical position (the Y coordinate of its top-left corner, can be from 1 to the height of the screen).

<Direction>HORIZONTAL_FIRST</Direction>

The direction in which records are inserted into the list (optional). Possible values are: "VERTICAL_FIRST" or "HORIZONTAL_FIRST" (default: VERTICAL_FIRST).

<RecordWidth>1</RecordWidth>

The width (in characters) of a single list record (i.e. cell). The default value is "1". For example, when wanting to repeat an occurrence every twenty characters (counting from the beginning of the previous record), enter 20 for this parameter and 0 for the value of the RecordHeight.

<RecordHeight>1</RecordHeight>;

The height (in characters) of a single list record (i.e. cell). Possible values can be in the range 1-99. The default value is "1". For example, when wanting to repeat an occurrence every two rows (counting from the first row of the previous occurrence), enter 2 for this parameter.

<RecordsNum>4</RecordsNum>

The total number of records to produce. Possible values can be in the range 1-99. The default value is "1".

Definition of a Loop Field

A loop field describes a repetitive element of the host screen, which is not a list. The most common example being the drawing of a frame of some sort which consists of simple (static) fields one below the other. Each simple field inside the loop field is repeated (cloned) either along the x-axis, y-axis or both ("table"). The distance between adjacent cloned fields is controllable, and it is the user's responsibility to make sure cloned fields are kept within the map's bounds.


```

<LoopField>
  <HorizontalClonesNum>1</HorizontalClonesNum>
  <HorizontalOffset>1</HorizontalOffset>
  <VerticalClonesNum>1</VerticalClonesNum>
  <VerticalOffset>1</VerticalOffset>
  <Field>
    ...
  </Field>
  <Field>
    ...
  </Field>
</LoopField>

```

Loop Field Parameters

<HorizontalClonesNum>1</HorizontalClonesNum>

The number of cloned fields (including the field itself) to be created in each row (optional, default is 1).

<HorizontalOffset>1</HorizontalOffset>

The number of columns between adjacent cloned fields (optional, default is 1 which means that there are no blank columns between cloned fields, just attributes).

<VerticalClonesNum>1</VerticalClonesNum>

The number of cloned fields (including the field itself) to be created in each column (optional, default is 1).

<VerticalOffset>1</VerticalOffset>

The number of rows between adjacent cloned fields (optional, default is 1 which means that there are no blank rows between cloned fields).

Defining Map Steps

The Map Steps defined here can be used in the Application Map.

```

<MapStep>
  <SendKeys>[ENTER]</SendKeys>
  <TargetScreen>ScreenName</TargetScreen>
  <CursorPos>
    <PosX>10</PosX>
    <PosY>5</PosY>
  </CursorPos>
  <InputField>
    <StartPos>
      <PosX>10</PosX>
      <PosY>5</PosY>
    </StartPos>
    <Text>input</Text>
  </InputField>
  <InputField>
    <StartPos>
      <PosX>10</PosX>
      <PosY>7</PosY>
    </StartPos>

```

```

    <GlobalVariableName>variableName</GlobalVariableName>
    <Text>prefix</Text>
</InputField>
<InputField>
    ...
</InputField>
<InputField>
    ...
</InputField>
</Mapstep>

```

Note:

1. It is only possible to place one map step from one specific screen to another specific target screen. When defining more than one, only the first one defined in the file will be displayed (e.g. if you can get from screen A to screen B both by sending the [enter] key or by sending the [PF5] key, the first map step defined is the one ApplinX uses).
2. It is not possible to define a map step from a screen to itself. Such map steps, if defined in the SDFX file, will be ignored.
3. Map step must contain a "SendKeys" element and a "TargetScreen" element. If one of them is missing, then the map step will not be added (a warning will be added to the server log).

Map Step Parameters**SendKeys**

The host key. Ensure that the host key you define is listed in the list of host keys which is displayed in the application properties, in the Keyboard Mapping tab, in the Host Key column.

For Example:

```
<SendKeys>[attn]</SendKeys>
```

CursorPos

The position of the cursor. For example there may be several map steps with the same [help] host key, each with a different target screen (the specific help menu) as long as they also have different cursor locations (the field for which the user wanted the help menu).

PosX

The column of the cursor position.

PosY

The row of the cursor position.

TargetScreen

The name of the destination screen, as defined in the destination screen's map definition in the same SDFX file.

InputField

Optional. Each map step can have zero or more input fields, each field represents input the user entered to a specific field when performing the map step.

StartPos

Determines the position on screen where the first character was entered.

PosX

The column of the cursor position.

PosY

The row of the cursor position.

GlobalVariableName

Optional. The name of the global variable whose value should be used as the text.

Text

The text entered.

Example of an SDFX File

```
<Mapset>
  <Name>Example</Name>
  <Version>1</Version>
  <Type>SIMPLE</Type>
  <Map>
    <Name>Example</Name>
    <PosX>1</PosX>
    <PosY>1</PosY>
    <Width>80</Width>
    <Height>24</Height>
    <CursorPosX>1</CursorPosX>
    <CursorPosY>1</CursorPosY>
    <IsPopup>>false</IsPopup>
    <Field class="SimpleField">
      <Length>10</Length>
      <Attribute>PROTECTED</Attribute>
      <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
      <Bright>NORM</Bright>
      <ForegroundColor>GREEN</ForegroundColor>
      <BackgroundColor>DEFAULT</BackgroundColor>
      <DbcsMode>NONE</DbcsMode>
      <HighlightBlink>>false</HighlightBlink>
      <HighlightReverse>>false</HighlightReverse>
      <HighlightUnderline>>false</HighlightUnderline>
      <OffsetFromLeft>0</OffsetFromLeft>
      <OffsetFromRight>0</OffsetFromRight>
      <OffsetFromTop>0</OffsetFromTop>
      <OffsetFromBottom>0</OffsetFromBottom>
      <Type>SIMPLE</Type>
      <Text>First Name</Text>
    <PosX>8</PosX>
```

```

    <PosY>3</PosY>
  </Field>
    <Field class="SimpleField">
      <Length>10</Length>
      <Attribute>PROTECTED</Attribute>
      <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
      <Bright>NORM</Bright>
      <ForegroundColor>WHITE</ForegroundColor>
      <BackgroundColor>DEFAULT</BackgroundColor>
      <Type>SIMPLE</Type>
      <Text>-----</Text>
      <PosX>8</PosX>
      <PosY>4</PosY>
  </Field>
    <Field class="SimpleField">
      <Length>10</Length>
      <Attribute>PROTECTED</Attribute>
      <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
      <Bright>NORM</Bright>
      <ForegroundColor>WHITE</ForegroundColor>
      <BackgroundColor>DEFAULT</BackgroundColor>
      <OffsetFromLeft>0</OffsetFromLeft>
      <OffsetFromRight>0</OffsetFromRight>
      <OffsetFromTop>0</OffsetFromTop>
      <OffsetFromBottom>0</OffsetFromBottom>
      <Type>SIMPLE</Type>
      <Text>-----</Text>
      <PosX>20</PosX>
      <PosY>4</PosY>
  </Field>
    <Field class="SimpleField">
      <Length>10</Length>
      <Attribute>PROTECTED</Attribute>
      <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
      <Bright>NORM</Bright>
      <ForegroundColor>BLUE</ForegroundColor>
      <BackgroundColor>DEFAULT</BackgroundColor>
      <DbcsMode>NONE</DbcsMode>
      <HighlightBlink>>false</HighlightBlink>
      <HighlightReverse>>false</HighlightReverse>
      <HighlightUnderline>>false</HighlightUnderline>
      <OffsetFromLeft>0</OffsetFromLeft>
      <OffsetFromRight>0</OffsetFromRight>
      <OffsetFromTop>0</OffsetFromTop>
      <OffsetFromBottom>0</OffsetFromBottom>
      <Type>SIMPLE</Type>
      <Text>Last Name</Text>
      <PosX>20</PosX>
      <PosY>3</PosY>
  </Field>
  <Field class="ListField">
    <Direction>HORIZONTAL_FIRST</Direction>
    <RecordWidth>1</RecordWidth>
    <RecordHeight>1</RecordHeight>
    <RecordsNum>15</RecordsNum>
    <Field>
      <Name>firstName</Name>
      <Length>10</Length>
      <Attribute>PROTECTED</Attribute>
      <DataType>DATA_TYPE_ALPHANUMERIC</DataType>
      <Bright>BRT</Bright>
      <DbcsMode>NONE</DbcsMode>

```

```
<Type>SIMPLE</Type>
<PosX>8</PosX>
<PosY>6</PosY>
</Field>
<Field>
  <Name>lastName</Name>
  <Length>10</Length>
  <Attribute>UNPROTECTED</Attribute>
  <DataType>DATA_TYPE_NUMERIC</DataType>
  <Bright>NORM</Bright>
  <ForegroundColor>WHITE</ForegroundColor>
  <BackgroundColor>DEFAULT</BackgroundColor>
  <Type>SIMPLE</Type>
  <PosX>20</PosX>
  <PosY>6</PosY>
</Field>
<Type>LIST</Type>
<PosX>1</PosX>
<PosY>1</PosY>
</Field>
<LoopField>
  <HorizontalClonesNum>1</HorizontalClonesNum>
  <HorizontalOffset>1</HorizontalOffset>
  <VerticalClonesNum>1</VerticalClonesNum>
  <VerticalOffset>1</VerticalOffset>
</LoopField>
</Map>
</Mapset>
```