

Development Guide



MiniApps XML format specification

Table of Contents

1	Introduction	3
2	Summary and definitions	4
3	Structure elements	6
3.1	Page	6
3.2	Title	7
3.3	Div	7
4	Auxiliary elements	9
4.1	Attributes	9
4.2	Attribute	9
4.3	Telegram usage	9
5	Navigation elements	11
5.1	Navigation	11
5.2	Link	11
5.3	Telegram keyboards	12
5.3.1	Limitations	13
5.3.2	Automatic keyboard structuring	13
5.3.3	Inline-keyboard	13
5.3.4	Special links	14
6	Text elements	15
6.1	Br	15
6.2	I	15
6.3	B	15
6.4	U	15
6.5	Ul	15
6.6	Ol	16
6.7	Li	16
6.8	A	16
6.9	Img	16
6.10	Telegram text formatting	17
7	Enhanced text elements	19
7.1	Meta:phone	19
7.2	Meta:mmclink	19
7.3	Meta:banner	19
8	Feedback elements	20
8.1	Select	20
8.2	Option	20
8.3	Input	20
9	Telegram attachments	22
9.1	Attachment sending	22
9.1.1	Parameters	22
9.2	Attachments receiving	25
9.2.1	Data types	26
10	Appendix 1. Page attributes	28
10.1	Payable and free pages attributes	28
10.2	Page identifier	28



1 Introduction

This document contains a description of XML documents format using for interaction of MiniApps platform with bots.

2 Summary and definitions

MiniApps page is a document, describing structure and content of service's resulting page. MiniApps page does not contain information about page's design. It only contains instructions for adaptation of page's information to particular design. MiniApps page is a XML-document which MiniApps receives from a bot. One MiniApps page contains description for mapping of one Service page.

Service page is a document sent to user's device. It is generated according specifics of client's software and mapped in accordance with MiniApps page content.

XML document may consist of the following elements:

1. [structure elements](#);
2. [auxiliary elements](#);
3. [text elements](#);
4. [navigation elements](#);
5. [enhanced text elements](#);
6. [feedback elements](#);
7. [attachments](#).

Structure elements include ones describing structure of resulting service page. Service page structure is determined by ordinal structuring and assigning administration attributes *id*, and *protocol*. Structure elements may nest all types of elements, including: *page*, *div* or *title*.

Administration attribute id is used to describe element's role in service page's structure. Using values of this attribute, the system can make various representations of current information block, depending on service design. Now the *id* attribute is not needed to describe USSD pages. USSD mapping for all service pages is performed regardless of *id* attribute values the next way: at the beginning of the page *title* headers are displayed in the order they appear, then *div* blocks followed by links. If a content provider wishes the mapping with considering of attributes values on USSD it must be negotiated separately. If it is not negotiated the attribute usage makes sense in case of WAP service version only is planned to be used. In this case the attribute values will be forwarded directly from a MiniApps page to a service one. It will allow the content provider to manage page appearance by means of CSS.

Administration attribute protocol is used for describing condition, whether current element will be displayed depending on the protocol used. If there is not the attribute, or it is empty, then current element will be displayed in any case.

The attribute values can be:

- **ussd** - an element will be displayed on USSD version only;
- **wap** - WML and XHTML-MP versions only;
- **telegram** - via Telegram messenger bots.

If an element should be displayed on different protocols, values are separated by space character. For instance *protocol="wap telegram"*.

Auxiliary elements include elements which contain additional information regarding a page and additional steps to be executed during mapping. For example billing, special instructions for application launching on user's device and so on. For details please see [Appendix 1](#). Auxiliary elements can allow nesting of only auxiliary elements, including *attributes* and *attribute*.

Navigation elements are elements, containing information about interrelations of service's pages. Using navigation elements, service's navigation menu is described. Navigation elements can allow nesting of *text* and *navigation* elements including *navigation* and *link*.

Text elements include elements describing information content of structure element and layout of this content in the context of the current information block. Text elements allow nesting of *text* and *enhanced text* elements including: *br*, *i*, *u*, *ol*, *ul*, *li*, *a*, *img*.

Enhanced text elements are elements implementing additional functionality of *text* elements. Name space of *the enhanced text* elements is: `xmlns:meta="http://whoisd.eyeline.com/sads/meta"`. Nesting of *text elements* is allowed. They include tags with "**meta:**" prefix. Content of the *enhanced text* elements may be changed.

Feedback elements are elements implementing feature of getting data from service users. Nesting of *feedback elements* is allowed. They include: **select**, **option** and **input**.

Attachments are used if some media content or location should be sent to a Telegram user or received from him.

3 Structure elements

3.1 Page

Path	/
Description	Document's root tag.
Occurrences	Document must contain only one <i>page</i> tag.

Attributes:

Name	Type	Assignment
<i>style</i>	text	Optional. It determines general scheme of external outlook representation of a service page. Depending on the value one MiniApps page can be represented different ways. If the attribute is not set than attribute value takes "default" one. It is used if a service requires special representation for a specific set of MiniApps pages. In this case the used values are negotiated with the support of this MiniApps installation. In case of <i>WAP</i> a value matches to different CSS styles.
<i>version</i>	text	Mandatory. Determines format's version of MiniApps page. Current version is 2.0.
<i>attributes</i>	text	Optional. This element can be used in case of using <i>telegram</i> as protocol. telegram.message.id - assigns unique identifier to a message. It is used to edit the message which has been sent before. telegram.message.edit - editing a message which has been sent before. The <i>telegram.message.id</i> must be specified also and the message with this identifier must be sent before in the same session. telegram.keep.session - do not close MiniApps session. It is useful if an inline-keyboard (see Link section) is used. For detailed description please see " Telegram usage " section.

Note:

MiniApps session:

- is linked to a service and a user;
- is created (if there is not) at the moment of push-message sending or user's message receiving;
- has limited lifetime.

MiniApps session will be closed if a message which does not require any answer was sent to a user. It means the message does not contain any input fields, links or inline-keyboard. *The telegram.keep.session forces the session be alive in this case.*

Example:

Listing	Explanation
<pre><page style="category"> ... </page></pre>	MiniApps page will be mapped in accordance with a style "category".

3.2 Title

Path	//title[parent:: name()="page"]
Description	Descendant of the <i>page</i> tag. It represents description of element's information title. Only text elements can be included in content.
Occurrences	Optional tag.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>protocol</i>	text	Optional, see administration attributes .
<i>attributes</i>	text	Optional. This element can be used in case of using <i>telegram</i> as protocol. For detailed description please see " Telegram usage " section.

3.3 Div

Path	//div[count(ancestor::div)!=0 or parent:: name()="page"]
Description	Descendant of <i>page</i> or <i>div</i> tag. It is a description of page's information block. Content may include <i>div</i> tags as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> ones.
Occurrences	Optional tag. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>protocol</i>	text	Optional, see administration attributes .
<i>type</i>	text	Optional. Specifies functional purpose of information block. Empty value – standard information block. <i>sms</i> – block's content should be send via SMS.
<i>attributes</i>	text	Optional. This element can be used in case of using <i>telegram</i> as protocol. For detailed description please see " Telegram usage " section.

Example:

Listing	Explanation
<pre> <page id="weather"> <div id="event"> Сегодня отличная<i>погода</i> </div> <div protocol="ussd"> + 23 C </div> <div protocol="wap" type="sms"> Температура: +23. Без осадков. Ветер южный. </div> </page> </pre>	<p>The page contains a weather column structure.</p> <p>The first div will be displayed via all protocols.</p> <p>The second div will appear on USSD only.</p> <p>Content of the third one will be send by SMS in case of using WAP.</p>

Note:

For correct working of USSD service all the service pages must contain some text content. It means it is not possible to use div tag with the attribute type="sms" if there is not any other text blocks or links displayed on USSD.

4 Auxiliary elements

4.1 Attributes

Path	/attributes
Description	Descendant of <i>page</i> tag. This is a section of auxiliary elements listing. The element contains only <i>attribute</i> tags.
Occurrences	Document contains no more than one tag. In case of empty tag or its absent it will be considered that there are no auxiliary elements indicated by service developers.

Attributes: Tag does not have XML attributes.

4.2 Attribute

Path	/attributes/attribute
Description	Descendant of <i>attributes</i> tag. Description of the auxiliary attribute. Does not have any content.
Occurrences	Optional. Can be contained by the <i>attributes</i> tag only. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>name</i>	text	Mandatory. Specifies unique identifier of the Auxiliary Attribute.
<i>value</i>	text	Mandatory. Specifies a value of the Auxiliary Attribute.

Example:

Listing	Explanation
<code><attributes /></code>	It is considered that there are no Auxiliary Attributes, specified by Service Developers.
<code><attributes> <attribute name="SADS-Content-type" value="SBT_1" /> <attribute name="resource-id" value="page1" /> </attributes></code>	It is considered that current SADS page will have two Auxiliary Parameters: SADS_Content_type=SBT_1 resource-id=page1

Note:

Possible attribute values are described in [Appendix 1](#).

4.3 Telegram usage

If *telegram* is used as the protocol attributes can be defined two ways:

1. as tags like described above;
2. as *attributes* parameters inside the next tags: **navigation**, **link** and **page**.

Example:

```

<page version="2.0" attributes="telegram.message.id: 42;
telegram.message.edit: true; telegram.keep.session: true">
  <div>Hello!</div>

  <navigation attributes="telegram.inline: true">
    <link pageId="_1.jsp"> 1 </link>
    <link pageId="_2.jsp"> 2 </link>
  </navigation>
</page>

```

The attributes can be specified both in an element and in nested tags. The next code blocks are equal:

```

<navigation attributes="telegram.inline: false; telegram.onetime: true">
  <link pageId="ok.jsp"> Ok </link>
</navigation>

```

```

<navigation attributes="telegram.onetime: true">
  <attribute name="telegram.inline" value="false"/>
  <link pageId="ok.jsp"> Ok </link>
</navigation>

```

```

<navigation>
  <attribute name="telegram.inline" value="false"/>
  <attribute name="telegram.onetime">true</attribute>
  <link pageId="ok.jsp"> Ok </link>
</navigation>

```

Attributes influence on nested elements if not redefined (like CSS). The next code blocks are equal:

```

<navigation attributes="telegram.inline: true">
  <link pageId="ok.jsp"> Ok </link>
  <link pageId="nope.jsp" attributes="telegram.inline: false"> Ok </link>
</navigation>

```

```

<navigation attributes="telegram.inline: true">
  <link pageId="ok.jsp" attributes="telegram.inline: true"> Ok </link>
  <link pageId="nope.jsp" attributes="telegram.inline: false"> Ok </link>
</navigation>

```

5 Navigation elements

5.1 Navigation

Path	/navigation
Description	Descendant of <i>page</i> tag. This is a section of <i>link</i> navigation elements listing. The element contains <i>link</i> tags only.
Occurrences	It is included by <i>page</i> tag. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>protocol</i>	text	Optional, see administration attributes .
<i>attributes</i>	text	Optional. This element can be used in case of using <i>telegram</i> as protocol. telegram.inline - inline-keyboard sending. For detailed description please see " Telegram usage " section.

Example:

Listing	Explanation
<pre><navigation> ... </navigation> <navigation protocol="ussd" > ... </navigation></pre>	<p>It is considered that navigation section will be displayed for all protocols if there is not <i>protocol</i> attribute. Additionally, section with <i>protocol="ussd"</i> attribute will be displayed for USSD protocol only.</p>

5.2 Link

Path	/navigation/link
Description	Descendant of <i>navigation</i> tag. It is a description of main menu link to another page. It can contain only <i>text</i> elements.
Occurrences	It is a mandatory tag inside a <i>navigation</i> tag. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>protocol</i>	text	Optional, see administration attributes .
<i>pageId</i>	text	Mandatory. It contains a page identifier to which the current link refers

Name	Type	Assignment
<i>accesskey</i>	number, #	Mandatory (except cases of using for form realization).It contains an identifier of a device's button by pressing which transfer by indicated link willbe accomplished (except cases of using for form realization).
<i>type</i>	text	Optional.Can be omitted by a service developer.Displays external representation of planned using of an element. back – return to the previous page. page – link to an another service page. <i>Page</i> type is assigned by default.
<i>priority</i>	[0-9]+	Optional.It is used for sorting.By default elements are displayed in order of determination.
<i>attributes</i>	text	Optional. This element can be used in case of using <i>telegram</i> as protocol. telegram.inline - inline-keyboard sending. telegram.external.url - link to an external address. It is used for inline-keyboard links. By a button clicking a user will be forwarded to an external link (out of Telegram chat) defined by the <i>pageId</i> attribute. For detailed description please see " Telegram usage " section.

Note:

- A page with no one link is a final page. If a user reaches a final page the session is ended.
- If there are links with attributes *telegram.inline: true* and they are not related to input fields (*input[navigationid]*) then an inline-keyboard will be generated from links with the attribute. Links without the attribute (if there are) will be ignored. It is a protocol limitation.

Example:

Listing	Explanation										
<pre><page version="2.0"> <div>...</div> <navigation attributes="telegram.inline: true"> <link pageId="kbd.jsp/1">1</link> <link pageId="kbd.jsp/2">2</link> <link pageId="kbd.jsp/3">3</link> <link pageId="kbd.jsp/4">4</link> <link pageId="kbd.jsp/5">5</link> </navigation> <navigation attributes="telegram.inline: true"> <link pageId="kbd.jsp/6">6</link> <link pageId="kbd.jsp/7">7</link> <link pageId="kbd.jsp/8">8</link> <link pageId="kbd.jsp/9">9</link> <link pageId="kbd.jsp/0">0</link> </navigation></page></pre>	<p>The next keyboard will be displayed:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>1</td><td>2</td><td>3</td><td>4</td><td>5</td> </tr> <tr> <td>6</td><td>7</td><td>8</td><td>9</td><td>0</td> </tr> </table>	1	2	3	4	5	6	7	8	9	0
1	2	3	4	5							
6	7	8	9	0							

5.3 Telegram keyboards

A keyboard is generated from navigation sets with links. Every navigation block matches a line of buttons (every *link* tag is a column). Below is a code which identifies three lines keyboard:

```
<page version="2.0">
  <navigation>
    <link pageId="kbd.jsp/1">1</link>
    <link pageId="kbd.jsp/2">2</link>
    <link pageId="kbd.jsp/3">3</link>
```

```

    <link pageId="kbd.jsp/4">4</link>
    <link pageId="kbd.jsp/5">5</link>
  </navigation>

  <navigation>
    <link pageId="kbd.jsp/6">6</link>
    <link pageId="kbd.jsp/7">7</link>
    <link pageId="kbd.jsp/8">8</link>
    <link pageId="kbd.jsp/9">9</link>
    <link pageId="kbd.jsp/0">0</link>
  </navigation>

  <navigation>
    <link pageId="cancel.jsp">Cancel</link>
  </navigation>

</page>

```

As a result a Telegram user will see a keyboard like this:

1	2	3	4	5
6	7	8	9	0
Cancel				

Every link must have *pageId* attribute identifying a page address on which the user will be forwarded.

5.3.1 Limitations

In accordance with a Telegram protocol a keyboard can have no more than 12 columns (extra columns will be ignored). In case of [inline-keyboards](#) this value is limited by 8 columns. A number of lines is unlimited. Vertical scrolling is used in case of many lines. If many columns are used button titles possibly will not be readable.

5.3.2 Automatic keyboard structuring

In case of keyboard line is too long for correct displaying on user's device MiniApps is able to organize a keyboard structure by itself.

Service configuration parameter	Default value	Description
<i>telegram.links.realignment.enabled</i>	false	If automatic structuring is enabled
<i>telegram.links.realignment.threshold</i>	10	Max line length. The line length is defined by symbols' number sum in every button located on this line.

In case of structuring is enabled and at least one line in an initial keyboard layout is longer than permitted length the initial layout will be ignored and the keyboard will be restructured to satisfy a max length requirement if possible.

5.3.3 Inline-keyboard

There is another type of keyboards called "*inline*". If used the keyboard will stay displayed after use and will be scrolled with other content.

Differences between inline-keyboard and usual one are:

1. it is linked to specific message but not to input field in a chat. As a result it will stay in a messages history independently on user's input and following pages;
2. it can be explicitly hidden / edited at any time;
3. clicking on links does not initiate sending of any text message, thus clicking information will not be saved in a chat history;
4. few inline-keyboards can exist in a chat history at the same time and all of them will be functional.

5.3.4 Special links

As address links special URLs are supported. By clicking the link a special action is performed:

Address (Pageld value)	Result
<i>telegram://request-contact</i>	By clicking a link a user is asked to send his contact (there is not a possibility to choose a contact from an address book). If there is a first contact request by this service of for this user a Telegram client will display a warning. See Attachment receiving
<i>telegram://request-location</i>	By clicking a link a user is asked to send his location. See Attachment receiving

6 Text elements

6.1 Br

Path	//br[count(ancestor::div ancestor::title ancestor::link)!=0]
Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements. Represents line break. Does not have any content.
Occurrences	Optional tag. Number is unlimited.

Attributes: There are not attributes.

6.2 I

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements. Represents italics lettering. Content contains text which should be displayed in italics. It is acceptable in case of WAP only. Nesting of text elements is acceptable.
Occurrences	Optional tag. Number is unlimited.

Attributes: There are not attributes.

6.3 B

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements. Represents bold lettering. Content contains text, which should be displayed in bold lettering. It is acceptable in case of WAP only. Nesting of text elements is acceptable.
Occurrences	Optional tag. Number is unlimited.

Attributes: There are not attributes.

6.4 U

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements. Represents underscored lettering. Content contains text, which should be displayed underscored. It is acceptable in case of WAP only. Nesting of text elements is acceptable.
Occurrences	Optional tag. Number is unlimited.

Attributes: There are not attributes.

6.5 Ul

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements. Represents content of bulleted list. Content contains <i>li</i> tags.
--------------------	---

Occurrences	Optional tag. Number is unlimited.
--------------------	------------------------------------

*Attributes:*There are not attributes.

6.6 OI

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements. Represents content of numbered list.Content contains <i>li</i> tags.
--------------------	--

Occurrences	Optional tag. Number is unlimited.
--------------------	------------------------------------

*Attributes:*There are not attributes.

6.7 Li

Description	Descendant of <i>ul</i> , <i>ol</i> .Represents content of list element.Content contains text of <i>list</i> element, also may contain <i>text</i> and <i>enhanced text</i> elements.
--------------------	---

Occurrences	Optional tag. Number is unlimited.
--------------------	------------------------------------

*Attributes:*There are not attributes.

6.8 A

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements.Represents content of a link not included in service's and page's navigation menu.It is acceptable in case of WAP only.Content contains displayed name of <i>link</i> . Also it can contain <i>text</i> elements.
--------------------	---

Occurrences	Optional tag. Number is unlimited.
--------------------	------------------------------------

Attributes:

Name	Type	Assignment
<i>href</i>	text	Specifies document's address to which a user should be transferred.
<i>title</i>	text	Optional. Adds a hint to link's text.

6.9 Img

Description	Descendant of <i>div</i> , <i>title</i> , <i>link</i> , as well as <i>text</i> and <i>enhanced text</i> elements and <i>feedback</i> elements.The element is used to display images on a service page.It is acceptable in case of WAP only.There is not any content.
--------------------	--

Occurrences	Optional tag. Number is unlimited.
--------------------	------------------------------------

Attributes:

Name	Type	Assignment
<i>alt</i>	text	Optional. Alternative text for image.

Name	Type	Assignment
<i>src</i>	text	Path to the image file.

Note:

1. Depending on a protocol, some *text* elements can be ignored. For example, all text attributes, except elements of a list and line break, will be ignored during structuring of USSD page or sending via SMS message.
2. *Text* elements do not have *id* and *protocol* attributes because they should be responsible for text layout of current information block only.
3. Tag is used inside *Link* tag as alternative to *imgurl* attribute in a former format.

6.10 Telegram text formatting

It is applicable if *telegram* protocol is used.

Text on a page can contain subset of HTML for layout. The next elements are applicable:

```

<b>bold</b>, <strong>bold</strong>

<i>italic</i>, <em>italic</em>

<a href="URL">inline URL</a>

<code>inline fixed-width code</code>

<pre>pre-formatted fixed-width code block</pre>

<br/>

```

Examples:

```

<page version="2.0">

  <div>
    Code block might be embedded like this:
    <br/>
    <code>
      &lt;code&gt;
        Code block
      &lt;/code&gt;
    </code>

    <br/>
    <br/>
    <br/>

    And a pre-formatted text:
    <pre>
      /\_/\
      (  o  o  )
      /  *  \
      \___/  \___/ meow!
      /  ___  \
      \  ___  /
    </pre>

```

```
</div>
</page>
```

```
<page version="2.0">

  <div>
    Bold text sample: <b>this</b> or <strong>that</strong>.
    <br/>
  </div>

  <div>
    Italic text sample: <i>this</i> and <em>that</em>.
    <br/>
  </div>

  <div>
    Inline URLs are also supported, let me attach one.
    <a href="http://google.com">Google</a>
  </div>

</page>
```

7 Enhanced text elements

7.1 Meta:phone

Description	Descendant of <i>div</i> and some of <i>text</i> elements.Specifies entity of phone number.Content contains phone number.
Occurrences	Optional tag. Number is unlimited.

*Attributes:*There are not attributes.

7.2 Meta:mmclink

Description	Descendant of <i>div</i> and some of <i>text</i> elements.Specifies a link to MMC resource.Does not have any content.
Occurrences	Optional tag. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>metainfo</i>	text	Specifies document's address, to which a user should go.
<i>data</i>	text	MMC resource identifier.
<i>allow</i>	text	Whether to create allowing record
<i>name</i>	text	Displayed name of resource
<i>extension</i>	text	Resource extension
<i>short</i>	text	If short links service be used

7.3 Meta:banner

Description	Descendant of <i>div</i> .It is used for including SharpAds banners into a service.Does not have any content.
Occurrences	Optional tag. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>target</i>	text	Comma-separated keywords of targeting

8 Feedback elements

8.1 Select

Description	Descendant of <i>div</i> tag. Description of single-choice list. Only <i>option</i> tags can be included in content.
Occurrences	Optional tag. It can be contained by <i>div</i> tag only. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>name</i>	text	Mandatory. Specifies name of parameter transferred to server.
<i>title</i>	text	Mandatory. Specifies invitation for choosing
<i>navigationId</i>	text	Mandatory. Defines navigation block identifier containing commands for working with a page form.

8.2 Option

Description	Descendant of <i>select</i> tag. Description of choice option. Content contains text of a choice and may contain <i>text</i> elements.
Occurrences	Optional. It can be contained by <i>select</i> tag only. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>value</i>	text	Mandatory. Specifies parameter value transferred to server.
<i>selected</i>	true false	Optional. WAP is supported only. Defines option selected by default. Default value is <i>false</i> .
<i>accesskey</i>	digit or #	Mandatory. Contains a button identifier by clicking on which the option will be selected

8.3 Input

Description	Descendant of <i>div</i> tag. It is an information input field. Content contains text of selection and may contain text elements.
Occurrences	Optional tag. It can be included by <i>select</i> tag only. Number is unlimited.

Attributes:

Name	Type	Assignment
<i>id</i>	text	Optional, see administration attributes .
<i>name</i>	text	Mandatory. Specifies a parameter value transferred to server.

Name	Type	Assignment
<i>value</i>	text	Mandatory. Defines field value by default.
<i>title</i>	text	Mandatory. Defines invitation to input.
<i>type</i>	text	Optional. Defines properties of input text. text – input of a text line hidden – hidden form parameter Number – input of numbers only (WAP) Password – password input (WAP only) Default value is <i>text</i> .
<i>navigationId</i>	text	Mandatory. Defines navigation block identifier containing commands for working with a page form.

Note: If necessary several forms can be specified on a page by setting up different information blocks for different feedback elements.

9 Telegram attachments

In case *telegram* protocol is used Telegram bot can send and receive attachments.

9.1 Attachment sending

To send a file to Telegram user a service page must have *attachment* element.

- The element must have a root location (outside of *div* or *navigation* blocks) anywhere in a page.
- One page can have few attachments.
- Pages can have attachments only or attachments, keyboard and text.

Example:

```
<page version="2.0">
  <div>Hi!</div>
  <div>Have two pictures:</div>
  <navigation>
    <link pageId="back.jsp">Назад</link>
  </navigation>

  <attachment type="photo" src="resources/cat.jpg"/>
  <attachment type="photo" src="resources/dog.jpg"/>
</page>
```

As a result of the example there will be three messages:

1. a message with the texts and keyboard;
2. the first file: "cat.jpg";
3. the second one: "dog.jpg";

The initial keyboard defined on the page will be duplicated while attachments sending. Thus the keyboard will stay on a screen after attachment is sent.

9.1.1 Parameters

Most of parameters for different attachments may vary.

Independent from the attachment type parameters														
Parameter	Mandatory	Value												
<i>type</i>	yes	Attachment type. Possible values: <table border="1" data-bbox="450 1697 1359 2018"> <thead> <tr> <th>Value</th> <th>Attachment type</th> </tr> </thead> <tbody> <tr> <td>video</td> <td>Video</td> </tr> <tr> <td>photo</td> <td>Photo</td> </tr> <tr> <td>voice</td> <td>Voice record</td> </tr> <tr> <td>document</td> <td>Arbitrary file</td> </tr> <tr> <td>audio</td> <td>Audio record</td> </tr> </tbody> </table>	Value	Attachment type	video	Video	photo	Photo	voice	Voice record	document	Arbitrary file	audio	Audio record
Value	Attachment type													
video	Video													
photo	Photo													
voice	Voice record													
document	Arbitrary file													
audio	Audio record													

		sticker location	Sticker Geo location
Attachment types are described below in the table.			
Parameters for attachments containing files			
<i>src</i>	yes	Sending document. Possible values:	
		Type	Example
		File identifier. If a file was uploaded to Telegram before the file identifier can be used.	AgADAgADmaoxG9KknwF4O978o3EMqb_EWSoABI5s-WWq46dqiR0AAgl
		Link. Relative or absolute link to a file or protocol - HTTP/HTTPS. Relative to the current page address. In this case a file will be downloaded to MiniApps at the moment of page treatment and then transmitted to Telegram.	resources/foo.jpg http://my-service.org/content.jsp?id=123
		BASE64 - Data URI document content.	data:image/gif;filename=test.txt;base64,./9j/4AAQSkZJRgABAQEASABIAAD== data:image/jpg;base64,./9j/4AAQSkZJRgABAQEASABIAAD==
<i>fileName</i>	no	File name. It is defined the next way: 1. If there is "fileName" parameter then it is used. 2. Otherwise "src" parameter is analyzed:	

		<ul style="list-style-type: none"> a. In case of link a file name from the link is used. b. If there is Data URI then file name is used in accordance with RFC 2397.
Attachment types		
Video		
<i>caption</i>	no	Message title in free form. Max length is 200 characters.
<i>src</i>	See src parameter above for details	
<i>fileName</i>	See fileName parameter above for details	
<i>duration</i>	no	Duration of video in seconds.
<i>width</i>	no	Video width
<i>height</i>	no	Video height
Photo		
<i>caption</i>	no	Message title in free form. Max length is 200 characters.
<i>src</i>	See src parameter above for details	
<i>fileName</i>	See fileName parameter above for details	
Voice record in accordance with ogg-file in OPUS format.		
<i>caption</i>	no	Message title in free form. Max length is 200 characters.
<i>src</i>	See src parameter above for details. Max voice file size is 50Mb.	
<i>fileName</i>	See fileName parameter above for details	
Arbitrary file		
<i>caption</i>	no	Message title in free form. Max length is 200 characters.
<i>src</i>	See src parameter above for details	
<i>fileName</i>	See fileName parameter above for details	
Audio record		
<i>caption</i>	no	Audio record title.
<i>src</i>	See src parameter above for details. Mp3 format is used.	
<i>fileName</i>	See fileName parameter above for details	
<i>duration</i>	no	Duration in seconds.
<i>performer</i>	no	Artist
Sticker		

<i>caption</i>	no	Message title in free form. Max length is 200 characters.
<i>src</i>		See src parameter above for details. Image in WebP format.
<i>fileName</i>		See fileName parameter above for details
Geo location		
<i>latitude</i>	yes	Latitude
<i>longitude</i>	yes	Longitude Example: <pre><page version="2.0"> <attachment type="location" latitude="55.008353" longitude="82.935733"/> </page></pre>

9.2 Attachments receiving

A Telegram user can send an attachments as well. In this case *input_type* parameter with the value *json* is present in the MiniApps answer. If there is, the *input* field or *bad_command* parameter should be interpreted as JSON array.

There is input field in the page:

```
<page version="2.0">
  <div>
    <input
      navigationId="submit"
      name="my_input"
      title="Input phone number or send contact"/>
    </div>
  <navigation id="submit">
    <link pageId="index.jsp">Done</link>
  </navigation>
</page>
```

If user has typed a text	User has sent an attachment
<p>The typed text is transmitted as a value of a parameter which name is the same as <i>input field</i> name.</p> <p>from the example above: <i>my_input=some user's input</i></p>	<p>User's attachment is sent as JSON array of "Input" type. Attachment indicator is also present.</p> <p><i>input_type=json</i> <i>my_input=JSON array</i></p> <p><i>Example:</i></p> <pre>[{ "type": "contact", "msisdn": "79131112233",</pre>

If user has typed a text	User has sent an attachment
	<pre> "name": "Andy" }, { "type": "location", "longitude": 82.935733, "latitude": 55.008353 }] </pre>

There is no an input field (there is not the input tag on the XML page):

If user has typed a text	User has sent an attachment
<p>The typed text is transmitted as a value of the parameter <i>bad_command</i>.</p> <p><i>bad_command</i>=some user's input</p>	<p><i>input_type</i>=json</p> <p><i>bad_command</i>=JSON array</p>

9.2.1 Data types

Field	Type	Mandatory	Description												
Input															
<i>type</i>	string	yes	Sending data type. Possible values: <table border="1" data-bbox="671 1111 1359 1361"> <thead> <tr> <th>Value</th> <th>Description</th> <th>Data type</th> </tr> </thead> <tbody> <tr> <td><i>contact</i></td> <td>Contact information - address book record</td> <td>InputContact</td> </tr> <tr> <td><i>location</i></td> <td>Geo location</td> <td>InputLocation</td> </tr> <tr> <td><i>file</i></td> <td>Some file</td> <td>InputFile</td> </tr> </tbody> </table>	Value	Description	Data type	<i>contact</i>	Contact information - address book record	InputContact	<i>location</i>	Geo location	InputLocation	<i>file</i>	Some file	InputFile
Value	Description	Data type													
<i>contact</i>	Contact information - address book record	InputContact													
<i>location</i>	Geo location	InputLocation													
<i>file</i>	Some file	InputFile													
InputContact															
<i>msisdn</i>	string	yes	Phone number from an address book												
<i>name</i>	string	yes	Name												
InputLocation															
<i>latitude</i>	floating point	yes	Latitude												
<i>longitude</i>	floating point	yes	Longitude												
InputFile															
<i>media_type</i>	string	yes	Attachment type. Possible values: <table border="1" data-bbox="671 1859 1359 2018"> <thead> <tr> <th>Value</th> <th>Attachment type</th> </tr> </thead> <tbody> <tr> <td>video</td> <td>Video</td> </tr> <tr> <td>photo</td> <td>Photo</td> </tr> </tbody> </table>	Value	Attachment type	video	Video	photo	Photo						
Value	Attachment type														
video	Video														
photo	Photo														

Field	Type	Mandatory	Description								
			<table border="1"> <tr> <td><i>voice</i></td> <td>Voice record</td> </tr> <tr> <td><i>document</i></td> <td>Arbitrary file</td> </tr> <tr> <td><i>audio</i></td> <td>Audio record</td> </tr> <tr> <td><i>sticker</i></td> <td>Sticker</td> </tr> </table>	<i>voice</i>	Voice record	<i>document</i>	Arbitrary file	<i>audio</i>	Audio record	<i>sticker</i>	Sticker
<i>voice</i>	Voice record										
<i>document</i>	Arbitrary file										
<i>audio</i>	Audio record										
<i>sticker</i>	Sticker										
<i>content_type</i>	string	no	MIME type of the attachment								
<i>size</i>	number	no	Attachment size in bytes								
<i>url</i>	string	yes	File link to upload								

10 Appendix 1. Page attributes

Current Appendix contains *auxiliary* page elements, their possible names, values and application field.

10.1 Payable and free pages attributes

Attribute should be specified in order to show that page's information content belongs to a specific content provider.

Billing attribute: name: *SADS-Content-Type* value: *pre-negotiated* Example: `<attribute name='SADS-Content-Type' value='SBT_CP_FREE' />` Page will be considered payable or free depending on the attribute value.

10.2 Page identifier

A content provider can receive reports of every service page delivery. The report contains subscriber's MSISDN and delivery status. For pages which delivery status is very important (like billing transaction completion or important information displaying) a private identifier can be set which will be specified for notification.

Name: *resource-id*

Value: any one.

Example: `<attribute name='resource-id' value='bill-transaction-12345' />`