



MARC Import/Edit Guidelines

Copyright © 2012 Inmagic (a subsidiary of SydneyPLUS International Library Systems). All rights reserved.

Inmagic®, the Inmagic logo, DB/Text®, DB/TextWorks®, BiblioTech®, and BiblioTech PRO® are registered trademarks; and Inmagic.net™, BibSpeed™, IntelliMagic™, and PowerPack™ are trademarks of Inmagic.

Other brand and product names are trademarks or registered trademarks of their respective holders. Use of any other product name does not imply endorsement of that product by Inmagic.

The information in this document is subject to change without notice and should not be construed as a commitment by Inmagic, which assumes no responsibility for any errors that may appear in this document.

WARRANTY

INMAGIC MAKES NO WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS. INMAGIC SHALL NOT BE LIABLE FOR ANY LOST PROFITS OR ANY OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN PARTICULAR, INMAGIC SHALL HAVE NO LIABILITY FOR ANY DATA OR PROGRAMS STORED OR USED WITH THIS PRODUCT, INCLUDING THE COSTS OF RECOVERING SUCH PROGRAMS OR DATA.

U.S. GOVERNMENT: If Licensee is acquiring the software on behalf of any unit or agency of the U.S. Government, the following shall apply:

(a) For units of the Department of Defense: RESTRICTED RIGHTS LEGEND: Use, duplication or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data Clause at DFARS 252.227-7013. (b) For any other unit or agency: NOTICE - Notwithstanding any other lease or license agreement that may pertain to, or accompanying the delivery of, the computer software and accompanying documentation, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Clause 52.227-19(c)(2) of the FAR.

Contractor/Manufacturer is Inmagic, 600 Unicorn Park Drive, Fourth Floor, Woburn, MA 01801, U.S.A.

Questions regarding any Inmagic product should be addressed to Inmagic or your local authorized Inmagic dealer.

Inmagic
600 Unicorn Park Drive
Fourth Floor
Woburn, MA 01801 U.S.A.
Telephone: 781-938-4444 or 800-229-8398
Fax: 781-938-4446
<http://www.inmagic.com>

support@inmagic.com

CustomerSvc@inmagic.com

sales@inmagic.com

wishlist@inmagic.com

- technical support questions
- general company, product, and services questions
- sales, product pricing, and custom solution questions
- feature requests

4.4.12

Contents

MARC Import	4
Installation	4
Configuring the Services	4
Configuring the Importer	5
Configuring Genie	5
Field Mapping	5
MARC File Upload	5
Troubleshooting	6
MARC Cataloging	7
Field Element	8
GenieMarcXmlField Element	9
MarcFields Element	9
GenieFields Element	10
Notes	10
User Interface	11
ControlParams Element	12
MarcGroupUserControlParams	14
MarcFieldParams	14

MARC Import

Genie v3.4 supports the ability to upload files in MARC Communications format, for import into the Catalog textbase. This requires installation of additional software needed to convert the information into a format that the Importer can handle.

Installation

The Inmagic Genie Installation Notes document includes a section describing how to install the Genie MARC Services software. Follow these instructions to install the services after installing Genie v3.4. We recommend that you specify the Genie application directory as the installation location, as the installed configuration files assume that folder structure. (If you install elsewhere, you may have to adjust the configuration files.)

Two services are installed. The MARC Background File Processor, installed in a folder named MarcBFP, handles the conversion from MARC Communications format to MARC 21 XML. The Background File Processor, installed in a folder named BFP, converts MARC 21 XML files into XML files suitable for import into the Catalog textbase, using mapping rules derived from information in a CONFIG file.

Genie MARC Services are started after they are installed.

Configuring the Services

Each service folder contains an AppSettings folder, and in that folder is an XML file containing configuration information for that service.

The default MarcBFP config file (MarcBFPCfg.xml) is as follows:

```
<MarcBFPCfg CheckInterval="10">
  <FileProcessSpec>
    <RequestFolder>..\MarcImport\MarcUploadRequest\</RequestFolder>
    <StagingFolder>..\MarcImport\MarcUploadStaging\</StagingFolder>
    <SinkFolder>..\MarcImport\MarcUploadSink\</SinkFolder>
    <ResultFolder>..\MarcImport\MarcXmlRequest\</ResultFolder>
    <ExceptionFolder>..\MarcImport\MarcUploadException\</ExceptionFolder>
  </FileProcessSpec>
</MarcBFPCfg>
```

The CheckInterval attribute on the MarcBFPCfg element specifies the time (in seconds) between checks for new files to process.

The RequestFolder element specifies the folder into which new MARC files will be dropped (by the Genie upload function) for processing and import.

The ResultFolder element specifies the directory where the MarcXML result of the conversion will be created – this will be the request folder for the next stage of processing.

The SinkFolder directory will contain the original MARC file, after processing is complete.

StagingFolder is an intermediate directory used during processing.

ExceptionFolder will receive files that fail to be processed for some reason (this functionality is not currently implemented).

As shown in the example above, the leading “..” notation can be used to designate the directory immediately above the MarcBFP directory. The default configuration file shown above references directories that are created when the Genie MARC services are installed. With the possible exception of the Request Folder, there should be no need to modify this file.

The default BFP config file (BFPCfg.xml) is as follows:

```
<BFPCfg CheckInterval="10">
  <FileProcessSpec>
    <RequestFolder>..\MarcImport\MarcXmlRequest\</RequestFolder>
    <StagingFolder>..\MarcImport\MarcXmlStaging\</StagingFolder>
    <SinkFolder>..\MarcImport\MarcXmlSink\</SinkFolder>
    <ResultFolder>..\MarcImport\GenieImportRequest\</ResultFolder>
  </FileProcessSpec>
</BFPCfg>
```

```

    <ExceptionFolder>..\MarcImport\MarcXmlException\</ExceptionFolder>
    <TransformFile>..\MarcImport\MarcToGenieImport.xsl</TransformFile>
  </FileProcessSpec>
</BFPCConfig>

```

The functionality is identical to the MarcBFP config file, with the addition of the TransformFile element. This specifies the XSLT transform that will be used to convert MarcXML format to Genie import format. A “standard” transform is installed with the Genie MARC services, and a new transform can be generated by Genie if the Genie Marc-related configuration file is modified.

The final output folder (ResultFolder) is the input to the Genie Importer.

Configuring the Importer

The appropriate Importer service for the WebPublisher platform in use must be installed and configured to import the files produced by the Background File Processor service into the Catalog textbase. The profile to set up for this import is as follows:

For the Catalog textbase, select Fielded Text Files.

- On the File Format tab:
 - Select **XML**.
 - Specify the default transform.
- On the Validation tab:
 - Select **Accept Overrides** (with or without **Update Validation Lists**).
- Import Directory: the full path and name of the GenieImportRequest folder

Move To: doesn't matter

Configuring Genie

Field Mapping

Genie is responsible for determining the field mapping between the MARC fields and the Genie fields, in other words, which MARC fields/subfields are copied to which Genie Catalog fields. This is done in (My)EditScreens.CONFIG. If MARC editing is enabled, this mapping is used when information on a MARC edit screen is saved. Whether or not MARC editing is enabled, this mapping determines which MARC information appears in which Genie field when MARC records are uploaded and imported.

An element in MyGenie.CONFIG drives the process by specifying where Genie will find the information it needs. This element names the edit screen config file, the edit screen definition in it, the transform that consumes the edit screen definition and emits the import transform, the name and location of that transform, and whether or not to retain XML MARC in the record itself (to support the ability to later extract the full MARC data). Example:

```

<MarcSupport MapFileName="ConfigFiles\MyEditScreens.config"
  MapScreenName="catalog_edit"
  MapScreenTransform="EditScreenToImportxsl.xsl"
  MapImportTransform="MarcImport\MarcToGenieImport.xsl"
  CreateCatMarcXml="true" />

```

Information in the next section (MARC Cataloging) explains how the field mapping is specified in MyEditScreens.config. If MARC import is supported in your configuration but MARC editing is not, the mapping can STILL be specified in an edit screen definition, but that definition would never drive an edit screen appearance if it doesn't have the default edit screen name ("catalog_edit").

MARC File Upload

This section describes how to set up Genie to support the upload of MARC Communications files.

Add the upload folder name to MyGenie.config. Open MyGenie.config and add this element as a child of the configurationsettings element. You will find a commented-out version in the v3.4 Genie.config file. If

you installed the Genie MARC Services someplace other than in the Genie installation folder, put the location and name of the MarcUploadRequest folder in the Folder attribute below. Use the fully qualified path if it is outside of the Genie folder structure.

```
<UploadFolders>
  <UploadFolder Name="UploadMARC"
                Folder="MarcImport\MarcUploadRequest"
                Textbase="Catalog" />
</UploadFolders>
```

Add an item to the navigation menu. Open MyNavBarMenu.config and add this code to an appropriate location. We recommend the "Other" menu, in place of or near the BookWhere Upload item.

```
<Item Id="Other_Import_Upload"
      Look-LeftIconUrl="marc.gif"
      Text="Upload File to Import"
      NavigateURL="bookwhere_upload.aspx ?Type=ImportFile &Name=UploadMARC
                  &EditScreen=import_upload "
      permission="Administrators Catalogers"></Item>
```

If you copy that element and paste it into your MyNavBarMenu.config file, be sure to remove the spaces in the NavigateURL value. They were inserted to make the text readable in this document.

Also note that the ampersands have to be encoded (& instead of &).

The Name parameter connects this menu item with the folder set up in MyGenie.config.

Copy the import_upload edit screen definition from the v3.4 EditScreens.config file to MyEditScreens.config. This is the definition. Change the text to suit your requirements.

```
<inmg:EditScreen Name="import_upload" TitleForAdd="Upload File to Import">
  <Toolbar Title="Data Entry Toolbar" ButtonCellwidth="260">
    <Button Name="Save" Text="Upload" />
  </Toolbar>
  <EditSection Title="Select File">
    <Field Style=". . .">%%StartTable</Field>
    <Field>%%StartRow</Field>
    <Field Display="Import File"
          HelpDisplayCss="sidemenutxt"
          UploadType="ImportFile"
          NewHelpText="Select your file . . ."
          CssClass="catalog_edit_textinput"
          UploadBoxSize="35">%%Fileupload</Field>
    <Field>%%EndRow</Field>
  </EditSection>
</inmg:EditScreen>
```

Troubleshooting

If one of the services (MarcBFP or BFP) detects an error, it will attempt to log it to a file named errors.log in its AppSettings directory. If this fails, it will log the error in the Application Event Log, accessible through the Computer Management Console (right-click My Computer and select Manage, then click on Event Viewer).

MARC Cataloging

Genie v3.4 has support for MARC cataloging. Bibliographic information is entered into MARC fields, complete with MARC tags, indicators, and subfields. The MARC information is retained in the record in a standard XML format, and the content is also copied into appropriate Genie fields, so that Genie searching and reporting work as usual.

This is achieved by adding MARC-specific user controls to a Catalog edit screen definition. The edit screen definition controls both the appearance of the MARC edit screen and the mapping of MARC fields into Genie fields.

The primary control is a MARC group. It will have a label, such as "Title", and one or more MARC fields associated with it. There may be an Add button or link, for requesting another field in the group or another instance of the current field.

Two sample edit screens with MARC support are included with the MARC module. One of them is MARC-centric, with MARC fields appearing in numerical order, each in its own group. The other is Genie-centric, with MARC fields contributing to a Genie field grouped together. For example, the 100 and 700 fields, both of which are mapped to the Author field, appear together in the group labeled "Author".

Part of the sample MARC-centric screen:

The screenshot shows a MARC-centric edit screen with four tabs at the top: "00x-1xx", "2xx-4xx" (selected), "5xx", and "6xx-8xx". Below the tabs are three groups of fields:

- * Title**: A text input field with a "245" tag, two indicator boxes, and a subfield box.
- Alternate Title**: A text input field with a "246" tag, two indicator boxes, and a subfield box. To the right of the subfield box are "+" and "-" buttons.
- Edition**: A text input field with a "250" tag, two indicator boxes, and a subfield box.

Part of the sample Genie-centric screen:

The screenshot shows a Genie-centric edit screen with four tabs at the top: "Biblio1" (selected), "Biblio2", "Serials", and "Leader/Control Fields". Below the tabs are four groups of fields, each with an "Add Tag" button:

- * Title**: An "Add Tag" button, a "245" tag, two indicator boxes, and a subfield box.
- Alternate Title**: An "Add Tag" button and a subfield box.
- Author**: An "Add Tag" button, a dropdown menu showing "100" (selected), two indicator boxes, a subfield box, and a "Remove" button.
- Corporate Author**: An "Add Tag" button, a dropdown menu showing "110" (selected), two indicator boxes, a subfield box, and a "Remove" button.

Field Element

The **Field** element has extra attributes and child elements that support MARC cataloging. The Type attribute has an additional value, "UserControl". If the Field type is UserControl, specify the name of the control in a new ControlName attribute. These are the options that support MARC cataloging:

- MarcGroupUserControl – handles display of the MARC field controls, specifies the Catalog field used to contain the MARC fields in XML format, and specifies the mapping of MARC to Genie fields
- MarcConsolidatorControl – handles mapping of MARC to Genie fields that cannot be managed by the group controls

Examples:

```
<Field Name="CatISBN" Display="ISBN" Type="UserControl"
  ControlName="MarcGroupUserControl">
```

```
<Field Name="XCatAll" Display=" " Type="UserControl"
  ControlName="MarcConsolidatorControl" ProcessLast="true">
```

The following table describes new values for two Field element attributes and two new attributes (shown with an asterisk). The two tables beneath it are more fully described on the next page.

Attribute	Explanation
Name	The AppName of the primary Genie field that the control will be populating, OR any unique name beginning with an X (a signal that prevents this control from participating directly in the construction of the Catalog record).
Type	Specify UserControl if using a MARC control.
ControlName *	Required if the Type is UserControl. Supported values: MarcGroupUserControl, MarcConsolidatorControl.
ProcessLast *	Optional, use with a MarcConsolidatorControl to specify that all other controls on the screen should be processed before this one.

A Field element that is a MARC group user control supports the following child elements.

Element	Required	Explanation
GenieMarcXmlField	Yes	The name of the field in the Catalog textbase that will contain the MARC XML.
MarcFields	Yes	Contains a child element for each MARC field in the control. These elements control the appearance of the specific MARC field controls.
GenieFields	No	Contains a GenieField element for each Genie field populated by the control. These elements control the mapping of the MARC fields in the control to Genie fields. They are not involved in the appearance of the edit screen.

A Field element that is a MARC consolidator control supports the following child elements.

Element	Required	Explanation
GenieMarcXmlField	Yes	The name of the field in the Catalog textbase that will contain the MARC XML.
GenieFields	Yes	Contains a GenieField element for each Genie field populated by the control. These elements control the mapping of the MARC fields in the control to Genie fields.

GenieMarcXmlField Element

This element specifies the name of the field in the Catalog textbase that will contain the MARC XML data populated by this control. In the sample MARC edit screens, the same field is used for all of the user controls on the edit screen. The ability to support multiple XML fields is available for future implementations.

MarcFields Element

The MarcFields element lists all of the MARC fields that can be entered or edited in this group control. A MARC field can be included in only one group control. Note that MARC fields that have not been included in any group control are not available for data entry. If it is important to include (and retain) information in MARC fields not mapped to Genie, include them in a group control but omit them from the mapping. This is likely to be the case with Leader and Control fields. If additional fields should be retained, signaling this in the user interface would help. (For example: the group control's Display parameter can communicate this information.)

The MarcFields element can contain one or more child elements from this list:

Element	Required	Use
Leader	No	Permits entry of MARC leader text.
ControlField	No	Use for MARC control fields (001 – 008).
DataField	At least one	Use for MARC data fields (010 - 999).

The Leader and ControlField elements have no indicator controls and expect no subfield markers. The DataField elements have indicator controls and can accept subfield markers. Note that any subfield can be entered in a MARC field (and retained), but only the subfields mapped by the GenieFields element will contribute information to the Genie fields accessed by the rest of the Genie application.

A Leader element has no content. Example:

```
<Leader />
```

ControlField and DataField elements contain a single MARC tag. Examples:

```
<ControlField>003</ControlField>
<DataField AddLinkVisible="false">250</DataField>
```

The following attributes can be used with these elements:

Attribute	Purpose
MultiLine	Set to "true" if the text typed in the textbox should wrap. The box will have a scrollbar. The default value is "false".
NumLines	Specify the height of the box in lines. The default value is 1. To set a higher number, be sure that MultiLine is also set to "true".

You can also use as attributes any of the style elements in the <ControlParams><MarcFieldParams> element, described later in this document.

GenieFields Element

This element specifies the mapping of the MARC fields and subfields to Genie fields. Its child element `Field` names the target Genie field, and there can be multiple child elements. Field elements have one or more child elements `Entry`, each of which creates a separate entry in the target Genie field. Entry elements have one or more child elements `MarcDataField`, which has a required attribute `Tag` identifying the contributing MARC tag. `MarcDataField` elements have one or more child elements `Subfield`, with required attribute `Code` (which specifies the subfield) and optional attribute `Delimiter` (the default value is the parameter set in `<ControlParams><MarcGroupUserControlParams><Delimiter>` if present, or new-line).

An example will help:

```
<MarcFields>
  <DataField . . .>245</DataField>
</MarcFields>
<GenieFields>
  <Field Name="CatTitle">
    <Entry>
      <MarcDataField Tag="245">
        <Subfield Code="a" Delimiter=" " />
      </MarcDataField>
    </Entry>
  </Field>
  <Field Name="CatSubtitle">
    <Entry>
      <MarcDataField Tag="245">
        <Subfield Code="b" Delimiter=" " />
      </MarcDataField>
    </Entry>
  </Field>
  <Field Name="CatResponsibility">
    <Entry>
      <MarcDataField Tag="245">
        <Subfield Code="c" Delimiter=" " />
      </MarcDataField>
    </Entry>
  </Field>
</GenieFields>
```

Notes

When a record on an edit screen is saved, Genie gets the content of each control on the edit screen, in no particular order, and assembles it into an XML document that will be sent to WebPublisher. If a control is populating more than one Genie field (a possibility with the MARC controls), the extra information is also saved. If two MARC fields are contributing information to the same Genie field, they should be in the same control, to avoid having the same Genie field included in separate places in the XML record. (If this problem exists, a Save will return a "Field exists more than once" message from WebPublisher.)

If they can't be in the same control and maintain the desired user interface (which is likely with the MARC-centric approach), use the `MarcConsolidatorControl` to handle the field mapping.

User Interface

A MARC field control consists of a MARC field tag, two boxes for indicators, and a text box for the contents of the field. The indicator boxes are absent for MARC control fields. The MARC field tag is in a read-only box if there is only one MARC field in the group. If there are multiple MARC fields in the group, there is a drop-down list from which the MARC field tag can be selected.

There are two basic styles available for supporting the ability to request another MARC field instance:

- Add/Remove buttons, with the Add button to the left of the MARC field control and the Remove button to the right
- Add/Remove links, both to the right of the MARC field control

There is only one Add button in a group, and it is always visible. The Remove button and the Add and Remove links disappear when the field is removed. For this reason, an attempt to remove the only control in a group using a Remove link will clear the control of any content, but leave it on the screen.

Note that there is no need to remove an empty MARC control.

This example illustrates a control using buttons. The MARC tag dropdown list permits selection of MARC tags 650 and 651.

Subjects

Add Field	650		0	aInternet (Computer network) zFlorida	Remove
	650		0	aLibrary information networks zFlorida	Remove
	650		0	aPublic libraries zFlorida xData processing	Remove

This example illustrates a pair of controls using links.

Subjects: Terms

650		0	aInternet (Computer network) zFlorida	+	-
650		0	aLibrary information networks zFlorida	+	-
650		0	aPublic libraries zFlorida xData processing	+	-

Subjects: Geographic Name

651				+	-
-----	--	--	--	---	---

ControlParams Element

This element is an optional child element of the EditScreen element. It provides default style parameters for the MARC controls within that edit screen. These parameters can be overridden for a specific control by using them as attributes on the affected control.

This example illustrates the parameters associated with an edit screen that uses Add/Remove links.

```
<ControlParams>
  <MarcGroupUserControlParams>
    <AddButtonVisible>false</AddButtonVisible>
    <AddButtonDisabled>false</AddButtonDisabled>
    <ScreenClass>catalog_marc_screen_white</ScreenClass>
    <Delimiter>\r\n</Delimiter>
    <MarcFieldParams>
      <IndicatorClass>catalog_marc_indicator</IndicatorClass>
      <TextBoxClass>catalog_marc_textinput</TextBoxClass>
      <TextBoxWidth>445</TextBoxWidth>
      <RemoveButtonVisible>false</RemoveButtonVisible>
      <AddLinkVisible>true</AddLinkVisible>
      <AddLinkText>+</AddLinkText>
      <AddLinkClass>catalog_marc_link</AddLinkClass>
      <RemoveLinkVisible>false</RemoveLinkVisible>
      <RemoveLinkText>-</RemoveLinkText>
      <RemoveLinkClass>catalog_marc_link</RemoveLinkClass>
    </MarcFieldParams>
  </MarcGroupUserControlParams>
</ControlParams>
```

This example shows the Add link hidden for a MARC 245 field, as this field is both required and non-repeating. (The Remove link is automatically hidden if the Add link is hidden.)

```
<DataField . . . AddLinkVisible="false">245</DataField>
```

Result:

★ Title

245	1	2	aA history of judicial intervention in church property disputes / cby Michael Thomas Pullin.
-----	---	---	--

This example illustrates the default parameters associated with an edit screen that uses Add/Remove buttons:

```
<ControlParams>
  <MarcGroupUserControlParams>
    <AddButtonVisible>true</AddButtonVisible>
    <AddButtonDisabled>>false</AddButtonDisabled>
    <AddButtonText>Add Tag</AddButtonText>
    <ScreenClass>catalog_marc_screen_color</ScreenClass>
    <ShowEmptyField>true</ShowEmptyField>
    <Delimiter>\r\n</Delimiter>
    <MarcFieldParams>
      <ShowEmptyField>true</ShowEmptyField>
      <IndicatorClass>catalog_marc_indicator</IndicatorClass>
      <TextBoxClass>catalog_marc_textinput</TextBoxClass>
      <TextBoxWidth>445</TextBoxWidth>
      <RemoveButtonVisible>true</RemoveButtonVisible>
      <RemoveButtonText>Remove</RemoveButtonText>
      <AddLinkVisible>>false</AddLinkVisible>
      <AddLinkText>+</AddLinkText>
      <RemoveLinkVisible>>false</RemoveLinkVisible>
      <RemoveLinkText>-</RemoveLinkText>
    </MarcFieldParams>
  </MarcGroupUserControlParams>
</ControlParams>
```

This example illustrates the style attributes for the required, nonrepeating 245 field. Note that the parameters outside of the <MarcFieldParams> element would be set in attributes applied to the <Field> element (with one exception, shown in the table below), while parameters defined as child elements of <MarcFieldParams> are set in attributes applied to the <DataField> element (or <Leader> or <ControlField>).

```
<Field width="100%" Name="CatTitle" Display="Title" Type="UserControl"
ControlName="MarcGroupUserControl" Required="true" AddButtonDisabled="true">
  <DataField . . . RemoveButtonVisible="false">245</DataField>
```

Result:

*** Title**

Add Field	245			
-----------	-----	--	--	--

MarcGroupUserControlParams

The following table lists the child elements that can be included within the <MarcGroupUserControlParams> element.

Element	Explanation
AddButtonVisible	Whether the Add button is visible, true/false, default is true.
AddButtonDisabled	Whether the Add button is visible but disabled, true/false, default is false.
AddButtonText	Text to appear on the Add button. The default is read from the global resource file: "Add Field".
Delimiter	The text that separates information already in a Genie field from new information added by another MARC field or another instance of the same MARC field. Override this parameter in the <Subfield> element (under <GenieFields><Field><Entry><MarcDataField>). Use the string "\r\n" for a new-line, which will force information into a separate entry in the Genie field.
ScreenClass	Name of the style in (My)Genie.css used for the group.
ShowEmptyField	Whether an empty field appears in a group that has no data yet; true/false, default is true.
MarcFieldParams	Contains parameters that affect fields added to the group control (see below).

MarcFieldParams

The following table lists the child elements that can be included within the <MarcFieldParams> element.

Element	Explanation
IndicatorClass	Name of the style in (My)Genie.css to use for the indicator boxes.
TextBoxClass	Name of the style in (My)Genie.css to use for the text box.
TextBoxwidth	Width of the text box in pixels. Default is 445.
AddLinkVisible	Whether the Add button is visible, true/false, default is true.
AddLinkText	Text to appear on the Add link. The default is "+".
AddLinkClass	Name of the style in (My)Genie.css to use for the link.
RemoveLinkVisible	Whether the Remove link is visible, true/false. Default is false. Note that if the Add link is not visible, the Remove link won't appear, either.
RemoveLinkText	Text to appear on the Remove link. The default is "-".
RemoveLinkClass	Name of the style in (My)Genie.css used for the link.
RemoveButtonVisible	Whether the Remove button is visible, true/false.

End of document