TogoDocClient plug-ins development manual

TogoDocClient Plug-ins development manual

Contents

1. The purpose of this document

This document, "TogoDocClient" which describes how to develop the software and plug-ins.

2. Developing TogoDocClient plug-ins

2.1. Building development environment

2.1.1. Install JDK 5.0

The following URL: Java SE Development Kit (JDK) 5.0 Download and install on your computer.

- http://java.sun.com/j2se/1.5.0/ja/download.html

2.1.2. Install Eclipse IDE

From the following URL Eclipse RCP and RAP Developers or Eclipse Classic download and unzip on your computer.

- http://java.sun.com/j2se/1.5.0/ja/download.html

In the subsequent description of Eclipse Helios (3.6) Packages of Eclipse Classic to use as explain.

2.1.3. Install target platform

From the following URL: to download the Eclipse SDK. This TogoDocClient that contains the files required for building.

- http://archive.eclipse.org/eclipse/downloads/drops/R-3.4.1-200809111700/.

However, the operating system you are using the current development of Windows XP, Vista, 7 Platform if any of those listed with the Windows, Mac OS X 10.5 or Mac OS X 10.6, if either Platform for Mac OSX (Mac / Carbon) have been described as a download. Any path to deploy the archive you downloaded. Since then, the path is denoted as $ {TDC_ECLIPSE}.

Second, be developing plug-ins you want to download TogoDocClient. However, the operating system you are using the current development of Windows XP, Vista, or if the version 7 Windows, Mac OS X 10.5 or Mac OS X 10.6, if any of Mac OSX (Mac / Carbon) Download version. Any path to deploy the archive you downloaded. Since then, the path is denoted as $ {TDC_TARGET}.

2.1.4. Configure target environment

Start Eclipse 2.1.2 deployed, Preferences dialog box opens. From the left side of the dialog "Plug-in Development> Target Platform", select the right section "Add ..." button you press.

"New Target Definition" dialog is opened when, "Nothing: ..." radio button is selected in the state, "Next>" to press. In the next screen, "Locations" tab "Add ..." button is depressed, "Add Content" dialog "Installation" select "Next" button pressing. Here "Location:" $ {TDC_ECLIPSE} to specify the path, "Finish" button to press.

Continued, "Locations" tab "Add ..." button is depressed, "Add Content" dialog "Installation" select "Next" button pressing. This "Location:" $ {TDC_TARGET} to specify the path, "Finish" button to press.

"Locations" tab of $ {TDC_ECLIPSE} and $ {TDC_TARGET} Once added, the same page "Environment", open to enter the information below.

- For Windows
  - ➢ Operating System: win32
  - ➢ Windowing System: win32
- For Mac OS X
  - ➢ Operating System: macosx
  - ➢ Windowing System: carbon
- Windows, Mac OS X Common
  - ➢ Architecture: x86

  - ➢ Execution Environment: J2SE-1.5

When you are finished over the top of the "Name:" text box "TogoDoc Client Plug-in" (Optional) Enter, "Finish" button to press. I return to the Preferences dialog, "TogoDoc Client Plug-in" to check if the "OK" button.

2.2. Prepare plug-ins development

In this case, TogoDocClient steps to extend a common, explains some of the steps.

2.2.1. Create project of the plug-ins

TogoDocClient is to extend the use of Eclipse plug-in mechanism. A project to develop a plug-in from the menu bar "File> New> Project ..." select the dialog that appears, "Plug-in Development> Plug-in Project" to create a select.

“New Plug-in Project” wizard's first page, enter the following information.

- Project Name
  - ➢ This project name

After entering the "Next>" and press the button, enter the following information on the next page.

- ID:
  - ➢ Same as this project name
- Version:
  - ➢ Version of this plug-in
- Name:
  - ➢ Display name of this plug-in
- Provider:
  - ➢ Name of the author or organization of this plug-ins
- Execution Environment:

  - J2SE-1.5

- Rich Client Application
  - No

Other items can be changed if necessary. After entering the above, "Finish" to create a project button.

2.2.2. Configure dependencies

Then plug TogoDocClient available to libraries, to set dependencies. A project created from the META-INF folder MANIFST.MF Open.

When you open MANIFEST.MF file in the editor, the bottom "Dependencies" tab of, "Required Plug-ins" section of the Add button. Enter the following information for each text box at the top of an open dialog, "OK" to set the dependency button.

- org.eclipse.core.expressions
- org.eclipse.core.resources
- org.eclipse.core.runtime
- org.eclipse.ui
- org.eclipse.ui.forms
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui

The above is an example, some settings are already included and what is actually possible, including those not available. In addition, jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui and the library is the core of TogoDocClient.

2.3. Extended used of Eclipse Commands

Many of the extensions are TogoDocClient, Eclipse can be made using the Komandofuremuwaku. The command can be added to the toolbar menu or by creating a handler for the command; you can run any program when a menu item is selected.

2.3.1. Create command

The command to create extensions of a plug-in. MANIFST.MF META-INF folder of the project file, even at the bottom of "Extensions" tab open. "All Extensions" section of "org.eclipse.ui.commands" If there is no item called, "Add" to add org.eclipse.ui.commands button is pressed.

Then, "All Extensions" section of "org.eclipse.ui.commands" Select an item from the context menu "New> command" is selected. Once the child elements are created automatically, select it in the "Extension Element Details" to edit the following section.

- id
  - Identifier of the command. Suppose, "com.example.tdc.exampleCommand" respectively.
- name
  - The name of the command. Suppose, "Sample Command" respectively.

In the above command "sample" command has been created named.

2.3.2. Create handler

The command handler in some way (eg, selected from the menu) when it is directed at a mechanism to launch the program it receives.

Program corresponding to the handler is created as a regular Java class, which inherits the base class “org.eclipse.core.commands.AbstractHandler”. Here is just a simple display handler-class of a message.

```
  package com.example.tdc;


  import org.eclipse.core.commands.AbstractHandler;

  import org.eclipse.core.commands.ExecutionEvent;

  import org.eclipse.core.commands.ExecutionException;

  import org.eclipse.jface.dialogs.MessageDialog;

  import org.eclipse.swt.widgets.Shell;

  import org.eclipse.ui.handlers.HandlerUtil;


  public class ExampleHandler extends AbstractHandler {


          public Object execute(ExecutionEvent event) throws ExecutionException {

                  Shell shell = HandlerUtil.getActiveShellChecked(event);

                  MessageDialog.openInformation(shell,

                                  "Sample handler",

                                  "This is a sample handler");

                  return null;

          }

}
```

Then, as the command handlers to register as extensions of a plug-in. MANIFST.MF META-INF folder of the project file, even at the bottom of "Extensions" tab open. "All Extensions" section of "org.eclipse.ui.handlers" If there is no item called, "Add" button to press “org.eclipse.ui.handlers” to add.

Then, "All Extensions" section of "org.eclipse.ui.handlers" Select an item from the context menu "New> handler" is selected. Once the child elements are created automatically, select it in the "Extension Element Details" to edit the following section.

- commandId
  - ➢ Command identifier of the target. 2.3.1, created by the command identifier.
- class
  - ➢ Handler program. Specify the name of the handler class that you created.

In the above command "sample" was registered for the handler. Thereafter, the command "sample" is indicated when a message dialog will be displayed.

2.3.3. Create menu items

Finally, we have created for the menu commands. MANIFST.MF META-INF folder of the project file, even at the bottom of "Extensions" tabs open. "All Extensions" section of "org.eclipse.ui.menus" If there is no item called, "Add" button to press org.eclipse.ui.menus to add.

Then, "All Extensions" section of "org.eclipse.ui.menus" Select an item from the context menu "New> menuContribution" is selected. Once the child elements are created automatically, select it in the "Extension Element Details" to edit the following section.

- locationURI
  - ➢ Identifier of the menu. Suppose "menu:tdc.ui.menu.help?after=additions" to enter.

This element specifies a menu of commands you want to add. In the example above is appended to the Help menu. For more information, refer to the TODO.

Next, specify the actual command. Select the element you just created from the context menu "New> command" is specified. Once the child elements are created automatically, select it in the "Extension Element Details" to edit the following section.

- commandId
  - ➢ The command identifier of the object. 2.3.1, created by the command identifier.

In addition, you can rewrite the label that appears as a menu item (usually using the name command), or you can specify an icon.

To save the current work after you have completed at least.

2.3.4. Create plug-ins

Finally, the format plug-ins can be distributed. MANIFST.MF META-INF folder of the project file, even at the bottom of "Overview" tab open. The tabs "Exporting" section "Export Wizard" link to select the name.

When you see the dialog, "Available Plug-ins ..." by checking that you created is selected plug-in, "Destination" tab "Directory" to specify the path to the destination. After entering the "Finish" button you press, the specified path "plugins" folder is created and named in the "[plug-in ID]_[version number ID].jar " file is created named.

To incorporate a plug-in created the above file TogoDocClient directly beneath the "plugins" directory that later, TogoDocClient may be started.

2.4. TogoDocClient Extended

We now introduce an extension TogoDocClient available.

2.4.1. Add menu bar items

To add to each menu item on the menu bar, use command framework introduced in 2.3. The menu you want to add "org.eclipse.ui.menus" on "menuContribution" can be specified.

Following items can be specified as a list of locationURI.

Target Menu	locationURI
File	menu:tdc.ui.menu.file?after=additions
TogoDoc	menu:tdc.ui.menu.dbcls?after=additions
Search	menu:tdc.ui.menu.search?after=additions

Target Menu	locationURI
Statistics	menu:tdc.ui.menu.statistics?after=additions
Window	menu:tdc.ui.menu.window?after=additions
Help	menu:tdc.ui.menu.help?after=additions

In addition, Mac OS X application menu to add a specific item can not be.

2.4.2. Add tool bar or action bar button

Even if the action bar to add buttons to a toolbar or use the command framework introduced in 2.3. The destination "org.eclipse.ui.menus" on "menuContribution" can be specified.

Following items can be specified as a list of locationURI.

Target Menu	locationURI
Toolbar	toolbar:tdc.ui.toolbar?after=additions
Article Explorer	toolbar:tdc.ui.articleexplorer?after=additions
Tag Explorer	toolbar:tdc.ui.tagexplorer?after=additions
Article Search	toolbar:tdc.ui.articlesearch?after=additions

2.4.3. Add context menu

Even if you add a variety of context menus, introduced in 2.3 command framework use. The menu you want to add "org.eclipse.ui.menus" on "menuContribution" can be specified.

Following items can be specified as a list of locationURI.

Target Menu	locationURI
Article Explorer	popup:tdc.ui.articleexplorer?after=additions
Tag Explorer	popup:tdc.ui.tagexplorer?after=additions
Create Tag Dialog	popup:tdc.ui.articleeditor.tagdialog?after=additions
Article Search	popup:tdc.ui.articlesearch?after=additions
List View	popup:tdc.ui.articleplaylist?after=additions

2.4.4. Add section for the article tab

When you add a section to document tabs, and create a section to add the class to update, create an instance of that class to create two classes Factory, Factory name of the class extension point element creatorFactory class Set.

Plugin.xml section to add information needed to describe is as follows.

Items	Content
extension point	jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui. articleSectionCreatorFactory
creatorFactory	Class implementing jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control. IArticleSectionCreatorFactory

2.4.4.1. Factory Class

Underlying the expansion of the class rather than creating a real section, which implements the interface below to generate the class Factory classes.

jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control.IArticleSectionCreatorFactory

Factory TogoDocClient instance of the class is generated only in one. Factory as an extension if you register more than one class instance is created for each extension one by one.

When opening new Article Tab, createSection method called, an instance of the generated class is created section.

2.4.4.2. Creating Section Class

We created a section on a tab, to view class information processing section of the literature. Article Tab new instance is created when opened, Article Tab instance when it is released to close. The following section is a generated class interface must implement.

jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control.IArticleSectionCreator

At instantiation is to be displayed without any documents, setArticle implementing documents to show what information was passed as a parameter when the method is called. setArticle method is called next time.

- Just opened a tab
- When it changes the document to be displayed
- The focused state when no tab is focused literature

Bibliography items that are not necessarily always contains a value, it must be implemented with consideration of the case does not contain.

2.4.5. Add column for the Article Search and List View

Article Search List View, and when you add a column to the table, adding columns to define a class, create an instance of that class to create two classes Factory, Factory name of the class extension point columnFactory Element is set to class.

Plugin.xml to add a column to describe the information required is as follows.

Item	Content

Item	Content
extension point	jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui. articleTableColumnFactory
columnFactory	Class implementing jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control. IArticleTableColumnFieldFactory

2.4.5.1. Factory Class

Underlying the expansion of the class rather than column defining, and the Factory class to produce a class must have the following interface to implement.

jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control.IArticleTableColumnFieldFactory

Factory PM4LS instance of Class only one is generated within the method is called immediately after createColumnFields, Factory instance of the class is no longer referenced. If you have registered as an extension to more than one class Factory, TogoDocClient Factory is recognized in the order by calling: createColumnFields Class, which brought together and used the returned list.

2.4.5.2. Column Definition Class

Both ArticleSearch and ListView are adding the column definition of the table. The contents are defined as those below.

- Column names (displayed)
- The initial width of the column
- The contents of the column
- Comparison of the column (used in sort)

Generate class instances are generated for each table column. ArticleSearch on the table and ListView the table of in and form a total of two times each generated. The column generated class must have the following interface implementation.

jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.control.IArticleTableColumnField

Bibliography items that are not necessarily always contains a value, it must be implemented with consideration of the case does not contain.

3. Developing TogoDocClient

3.1. Building the development environment

3.1.1. Install JDK 5.0

The following URL: Java SE Development Kit (JDK) 5.0 Download and install on your computer.

- http://java.sun.com/j2se/1.5.0/ja/download.html

3.1.2. Install Eclipse IDE

From the following URL Eclipse Classic download and unzip on your computer. After this a destination, $ {ECLIPSE} referred to.

- http://www.eclipse.org/downloads/

In the subsequent description of the Eclipse Helios (3.6) Packages of Eclipse Classic to use as explain. In addition, Eclipse workspace path to the startup $ {WORKSPACE} shall be denoted.

3.1.3. Install SVN Client

At 3.1.2, was developed in Eclipse, launch from the menu bar "Help> Install New Software ..." is selected. Install dialog box appears, "Work with:" drop-down "Helios - ..." is selected.

Because the list of installable software for a while, "Collaboration" to select from the following two items.

- Subversive SVN Team Provider

- Subversive SVN JDT Ignore Extensions

When you select a point or two, "Next>" button is depressed, follow the steps to install the software wizard. Asked to restart so that the installation is complete, follow the instructions to restart Eclipse.

After restarting from the menu bar "Window> Open Perspective> Other..." Select more "SVN Repository Exploring" is selected.

Here, "Install Connectors" dialog will pop up so that, "SVN Kit" only begins to select "Finish" button to press. Thereafter, follow the steps in the wizard after software install, follow the instructions to restart Eclipse.

After restarting, follow these steps: Preferences dialog is displayed.

- Windows
  - ➢ From the menu bar "Window> Preferences" is selected
- Mac OS X
  - ➢ From the menu bar "Eclipse> Preferences" is selected

Preferences dialog box from the left of the "Team> SVN", select the right "SVN Connector" tab open. "SVN Connector:" drop-down box to select the latest version, "OK" button to press.

3.1.4. Check out source code from the repository

From the menu bar "Window> Open Perspective> Other..." Select more "SVN Repository Exploring" is selected. Left of the "SVN Repositories" view the action bar (bottom right or tab) from "New Repository Location" and press the button, the dialog "General" tab, enter the following information.

- URL: http://svn.sourceforge.jp/svnroot/togodoc/client/tags/[RevisionNumber]
- User: (SVN's User Name)
- Password: (SVN's Password)

When added to the repository, select the following folders from the context menu "Check Out" is selected.

- jp.ac.u_tokyo.k.cb.wiwasaki.tdc
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc-feature
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc-site
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.analyzer.jpedal
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.analyzer.jpedal-feature
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.analyzer.jpedal.nl_ja
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.base
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.base-feature
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.build
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core.nl_ja
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core.test
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.carbon
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.nl_ja
- jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.test
- tdc-documents
- tdc-releases

However, jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui.carbon is not checked out Windows environment. Mac OS X, which can only be built under the project.

With the check-out is completed, should have listed several projects as an error.

3.1.5. Install target platform

From the following URL Eclipse SDK download. This TogoDocClient that contains the files required for building.

- http://archive.eclipse.org/eclipse/downloads/drops/R-3.4.1-200809111700/

However, the operating system you are using the current development of Windows XP, Vista, 7 Platform if any of those listed with the Windows, Mac OS X 10.5 or Mac OS X 10.6, if either Platform for Mac OSX (Mac / Carbon) have been described as a download. Any path to deploy the archive you downloaded. Since then, the path is denoted as $ {TARGET}.

Then, from the Delta Pack download page. This, TogoDocClient is needed to build a variety of platforms. After downloading, expand the archive, it contains the following plugins folder contains all of the .jar files to ${TARGET}/plugins copy below.

Finally, you just get ${WORKSPACE}/tdc-releases/rcp.nl_ja included below all .jar file ${TARGET}/plugins copy below.

3.1.6. Configure target environment

Start Eclipse 3.1.2 was deployed, Preferences dialog box opens. From the left side of the dialog "Plug-in Development> Target Platform", select the right section "Add ..." button you press.

"New Target Definition" dialog is opened when, "Nothing: ..." radio button is selected in the state, "Next>" to press. In the next screen, "Locations" tab "Add ..." button is depressed, "Add Content" dialog "Installation" select "Next" button pressing. Here "Location:" ${TARGET} to specify the path, "Finish" button to press.

"Locations" tab is added when the ${TARGET}, see the "Environment", open to enter the information below.

- For Windows
  - ➢ Operating System: win32
  - ➢ Windowing System: win32
- For Mac OS X
  - ➢ Operating System: macosx
  - ➢ Windowing System: carbon
- Windows, Mac OS X common

- Architecture: x86
- Execution Environment: J2SE-1.5

When you are finished over the top of the "Name:" text box "TogoDocClient" (optional) enter, "Finish" button to press. I return to the Preferences dialog, "TogoDocClient" to check if the "OK" button.

After the descriptions above, all projects is performed clean and build.

3.2. Building TogoDoc Client

Start Eclipse 3.1.2 was deployed, jp.ac.u_tokyo.k.cb.wiwasaki.tdc.build scripts to expand the project folder. Build.sample.properties folder contains the same directory as the file "build.properties" to copy the name.

Copy the file is opened, the settings for the build "set name=value" have been listed in the format. Modify the environment to suit the following pairs.

Name	The value to be set
version	Build version
buildDirectory	Building working directory
base	Parent directory of ${TARGET}
baseLocation	${TARGET}
baseos	win32 or macosx
basews	win32 or carbon

After saving your files or settings from the file context menu "Run As> Ant Build" is selected. After execution of Ant build script, tdc-releases and the release should refresh the project has generated a binary version of the specified folder.

Similarly, jp.ac.u_tokyo.k.cb.wiwasaki.tdc-site and refresh the project, features and plugins should have been generated files in a directory named for the update site. When building an update site, site.xml along with features, plugins folder may be placed online.

3.3. Debugging TogoDoc Client

3.3.1. Start TogoDoc Client from IDE

tdc.product project beneath jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui select the file from the context menu "Debug As> Eclipse Application" selected. Then, TogoDocClient start under the control of the development environment.

If you need more detailed logs, the menu bar "Run> Debug Configurations ..." is selected. Dialog opens, the left side of "Eclipse Application> tdc.product" Choose, Tracing change the contents of the tab as follows.

- Check Enable tracing
- Select jp.ac.u_tokyo.k.cb.wiwasaki.tdc.core, debug check
- Select jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui, debug check

After the settings above, Debug and you press a button to start TogoDocClient.

3.3.2. Attach TogoDoc Client from debugger

TogoDocClient started outside of the development environment when you attach a debugger, after making the specified configuration file for the debugger you need to start TogoDocClient. TogoDocClient configuration files are located below.

- For Windows
  - ➢ <Destination>/TogoDocClient.ini
- For Mac OS X
  - ➢ <Destination>/TogoDocClient.app/Contents/MacOS/TogoDocClient.ini

Open the file, the end of the file (or later than the line -vmargs) Add the following line.

```
-Xdebug
```

```
-Xnoagent

-Djava.compiler=NONE

-Xrunjdwp:transport=dt_socket,server=y,suspend=y,address=8000
```

Save the file and the configuration is complete, TogoDocClient started. In this configuration, TCP accepts the 8000 port to attach debug, the program does not start until it is attached.

In the development environment, debug by attaching to the process. The menu bar "Run As> Debug Configurations ..." Open the left from the "Remote Java Application" is selected in the context menu "New" is selected. The resulting screen, make the following settings.

- Name: (Free)
- Project: jp.ac.u_tokyo.k.cb.wiwasaki.tdc.ui
- Host: localhost
- Port: 8000

In the above configuration "Debug" button is pressed and had to wait to attach the debugger starts TogoDocClient start.

3.3.3. Logging detail messages

TogoDocClient produce a detailed log of when using 3.3.2, open the file described in the beginning of the file (or sooner than the line -vmargs) to add the following line.

```
-debug
```

After saving the configuration file when you start TogoDocClient behind the scenes, a variety of logs will be generated. This log is generated on each of the following directories.

- Windows
  - <Destination>/workspace/.metadata
- Mac OS X

- <Destination>/TogoDocClient.app/Contents/MacOS/workspace/.metadata

3.3.4. Configure update site

TogoDocClient to test software updates before you build a site update, open the file described in 3.3.2, at the end of the file (or later than the line -vmargs) Add the following line.

```
-Dtdc.update.site=<Update Site URL>
```

The update site URL "file:" what can be the beginning. For example, if a local jp.ac.u_tokyo.k.cb.wiwasaki.tdc-site path to the project will be available from the update site the result is always the final build.

The test site is updated 3.3.1 from the development environment as TogoDocClient might go wrong if you start.

3.3.5. Display English message resources

If you start as an English TogoDocClient, open the file you described 3.3.2, the beginning of the file (or sooner than the line -vmargs) to add the following line.

```
-nl en
```

TogoDocClient will automatically switch to the original language of the operating system language setting. If you start Japanese version from outside of the Japanese, change "en" to "ja" to start to convert.

3.4. Release TogoDoc Client

3.4.1. Adjust Versioning number

To release, all plug-ins, features, it is necessary for pre-and product version changes. If you have conflicting versions could cause a problem with using the update site update.

In order to build the update site, jp.ac.u_tokyo.k.cb.wiwasaki.tdc-site project's site.xml file, add a feature to do a release.

3.4.2. Runnable package

The executable package, using 3.2 as it is generated by the archive file.

3.4.3. Update site

Update site, 3.2 jp.ac.u_tokyo.k.cb.wiwasaki.tdc-site generated by the project plugins folder content, features the contents of the folder and use the site.xml file.

How do I update the site over the content of those current. Also, if you update your site from scratch, site.xml of features to be included, that you remove the entry for the version that does not exist.

3.4.4. DTD repository

TogoDocClient is currently communicating with the server are using the XML DTD, and some sites have been identified DTD is incorrect distribution. To avoid such problems and now use the DTD to make another offer from the server.

- http://togodoc.sourceforge.jp/client/dtd/

When added to the new DTD, the DTD file name without changing the URL above, be placed immediately below.

3.4.5. Setting SVN

After all work to create a release tag. The entire contents of the tag on the trunk release, the name of the release version.