



WFSTDМ

外部仕様書

外部仕様書	プログラム名	WFSTDM	page	
			4	
	ページタイトル		Doc-No	

1. はじめに

本書は WFSTDM の外部仕様について記載した文書である。


外部仕様書	プログラム名	WFSTDM	page	
			5	
	ページタイトル		Doc-No	

2. WFSTDM コマンド

2-1. 形態素モデル FST 作成

形態素モデル FST 作成コマンドを使用して、辞書と 3-gram のデータから形態素モデル FST を作成することができる。


書式	make_mphfst lex 3gram morph.fst	
引数	lex	辞書ファイルを指定する。
	3gram	3・gram ファイルを指定する。
	morph.fst	出力する形態素モデル FST を指定する。
戻り値	0 : 正常終了 1 : 異常終了	
説明	lex, 3gram で指定された辞書、3・gram のデータから形態素モデル FST morph.fst を作成する。	
使用例	<pre>\$ make_mphfst btec+hosei+vt_usv+propn124K+name+cook+plane+mailer_com.v11003j.v111219.lex ¥ btec+hosei+vt_usv+propn124K+name+cook+plane+mailer_com.v11003j.v111219.3gram morph.fst # 辞書ファイルとして btec+hosei+vt_usv+propn124K+name+cook+plane+mailer_com.v11003j.v111219.lex を指定 # 3・gram ファイルとして btec+hosei+vt_usv+propn124K+name+cook+plane+mailer_com.v11003j.v111219.3gram を指定 # 形態素モデル FST の出力先として morph.fst を指定</pre>	
備考	・ 使用するためには、make_mphfst コマンドを配置している bin(例 : /export/MCML/dmplatform/wfstdm/build/bin)にパスを通す必要がある ⇒make_mphfst コマンドでbinディレクトリ内の他のコマンドをパス指定なしで使用しているため	

外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		6	
			Doc-No	

2-2. 言語理解 FST 作成


言語理解 FST 作成コマンドを使用して言語理解 XML から FST を作成することができる。

書式	make_slufst [-cfh -l morph.fst -t lang_code] slu.xml slu.fst	
オプション	-c	フレーズのクロージャーループを作成する。
	-f	concept による出力タグを発話の先頭に出力する。
	-h	ヘルプ(コマンドの使い方)を出力する。
	-l morph.fst	形態素解析時に使用するモデル(morph.fst)を指定する。
	-t lang_code	言語コード(ISO 639-1)を指定する。
引数	slu.xml	言語理解 XML を指定する。
	slu.fst	出力する言語理解 FST を指定する。
戻り値	0 : 正常終了 1 : 出力する言語理解 FST が指定されていない	
説明	言語理解 XML から FST を作成する。	
使用例	<pre>\$ make_slufst -l/export/MCML/dmbuilder/wfstdm/morph/CookingMorph.bin.fst slu/cooking.xml cooking_slu.bin.fst # 形態素解析モデルとして CookingMorph.bin.fst を指定 # 言語理解 XML として cooking.xml を指定 # 言語理解 FST の出力先として cooking_slu.bin.fst を指定</pre>	
備考		


外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		7	
			Doc-No	

2-3. WFSTDM プロジェクト作成

書式	make_project [OPTION]... [KEY=VALUE]...	
オプション	--project=project	【必須】 作成するプロジェクト名を指定する。
	--language=language	【必須】 言語(ja, en, zh のいずれか)を指定する。
	--wfstdm_home=wfstdm_home	WFSTDM_HOME を指定する。省略した場合は環境変数\$WFSTDM_HOME を使用し、これが設定されていない場合はデフォルト(/export/MCML/dmplatform/wfstdm)を使用。
	--sample=sample_project	サンプルとして使用するプロジェクトを指定する。
	--help	本コマンドの使用方法を出力する。
	--verbose	詳細ログを出力する。
戻り値	0 : 正常終了 1 : 入力エラー(プロジェクト名、言語の指定に誤りあり) 2 : WFSTDM_HOME 配下にサンプルプロジェクトを管理するディレクトリが存在しない 3 : 指定されたサンプルプロジェクトが存在しない 4 : 指定されたプロジェクト名が既に存在する 5 : 指定されたプロジェクト名と一致するディレクトリが存在する	
説明	WFSTDM プロジェクトを作成する。 サンプルプロジェクトを使用してプロジェクトを作成することが可能。 \$WFSTDM_HOME/template/project 配下のプロジェクトをサンプルプロジェクトとすることができる。	
使用例	#DMBuilder のユーザー管理ディレクトリへ移動 \$ cd /export/MCML/dmplatform/dmbuilder/user/nict	


外部仕様書	プログラム名	WFSTDM	page	
			8	
	ページタイトル		Doc-No	

	<pre># サンプルプロジェクトを確認 \$ls/export/MCML/dmplatform/wfstdm/template/project cooking</pre> <pre># サンプルプロジェクト cooking を使用して日本語のプロジェクト my_cooking を作成 \$ make_project --project=my_cooking --language=ja --sample=cooking</pre>
備考	<p>➤ DMBuilder との共有</p> <p>作成したプロジェクトを DMBuilder と共有するため、必ず本コマンドは DMBuilder のユーザー管理ディレクトリ (/export/MCML/dmplatform/dmbuilder/user/[ユーザーID]) 配下で実行すること。</p>

外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		9	
			Doc-No	


2-4. アクションコードコンバーター

書式	compile_task [OPTION]... action.cpp	
オプション	-a dm_main.cpp	action.cpp とアペンドする対話メインファイルを指定する。
	-h output.hpp	action.cpp から C++ヘッダーファイル output.hpp を出力する。
	-c output.cpp	action.cpp から C++ソースファイル output.cpp を出力する。 -a オプションが指定されている場合は、対話メインファイルのコードも出力する。
	-o output.cpp	action.cpp から C++ソースファイル output.cpp(ヘッダー情報も含む)を出力する。 -a オプションが指定されている場合は、対話メインファイルのコードも出力する。
引数	action.cpp	C++ファイルへコンバートするアクションコードを指定する。
戻り値	0 : 正常終了 1 : 異常終了(オプション、引数指定の不正、またはアクションコードの文法エラー)	
説明	アクションコードをコンバートし、C++ファイルを出力する。	
使用例	# アクションコードに対話メインファイルをアペンドし、C++ソースファイル(ヘッダー情報も含む)cooking_code.cpp を出力 \$ pwd /export/MCML/dmplatform/dmbuilder/user/nict-wfstdm/cooking_ja \$ compile_task -a /export/MCML/dmplatform/wfstdm/build/src/dm_main.cpp -o cooking_code.cpp action.cpp	
備考		


外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		10	
			Doc-No	

2-5. WFSTDM ビルド

書式	dm_make [OPTION]... [KEY=VALUE]...	
オプション	--help	本コマンドの使用方法を出力する。
	--language=language	【必須】言語(ja, en, zh のいずれか)を指定する。
	--morph=morph	形態素モデルを指定する。省略した場合はデフォルトを使用。 デフォルト) 日本語 : /export/MCML/dmplatform/wfstdm/morph/morph_ja.bin.fst 英語 : /export/MCML/dmplatform/wfstdm/morph/morph_en.bin.fst 中国語 : /export/MCML/dmplatform/wfstdm/morph/morph_zh.bin.fst
	--wfstdm_home=wfstdm_home	WFSTDM ホームを指定する。省略した場合はデフォルトを使用。デフォルト) /export/MCML/dmplatform/wfstdm
	--standalone	スタンドアロンモードでビルドする。
	--project=project	プロジェクトを指定する。省略した場合はカレントディレクトリをプロジェクトとする。
	--only_action	アクションコードのみをビルドする。
	--verbose	ビルドに関する詳細ログを出力する。
戻り値	0 : 正常終了 1 : 指定したプロジェクトが適切でない 2 : 指定した言語が適切でない、または指定されていない 3 : 指定した形態素モデルが適切でない、または存在しない 4 : 指定した WFSTDM ホームが適切でない、または存在しない	


外部仕様書	プログラム名	WFSTDM	page	
			11	
	ページタイトル		Doc-No	

	<p>5 : 指定したプロジェクトが存在しない</p> <p>6 : シナリオ WFST のコンパイルに失敗</p> <p>7 : WFSTDM のビルドに失敗</p> <p>8 : 言語理解 FST の作成に失敗</p> <p>9 : 対話シナリオ WFST のコンポーズに失敗</p>
説明	<p>WFSTDM をビルドする。</p> <p>ビルド処理は dm_build コマンドで実施している。</p>
使用例	<p># スタンドアロンモードでビルド</p> <pre>\$ dm_make --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking --language=ja --standalone</pre> <p># MCML サーバーシステムモードでビルド</p> <pre>\$ dm_make --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking --language=ja</pre> <p># アクションコードのみビルド</p> <pre>\$ dm_make --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking --only_action</pre>
備考	

外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		12	
			Doc-No	

2-6. WFSTDM 起動


書式	wfstdm start [OPTION]... [KEY=VALUE]...	
オプション	--project=project	処理対象となる WFSTDM のプロジェクトを指定する。省略した場合はカレントディレクトリをプロジェクトとする。
	--language=language	【必須】言語(ja, en, zh のいずれか)を指定する。
	--morph=morph	形態素モデルを指定する。省略した場合はデフォルトを使用。 デフォルト) 日本語 : /export/MCML/dmplatform/wfstdm/morph/morph_ja.bin.fst 英語 : /export/MCML/dmplatform/wfstdm/morph/morph_en.bin.fst 中国語 : /export/MCML/dmplatform/wfstdm/morph/morph_zh.bin.fst
	--standalone	スタンドアロンモードで起動する。
	--port=port	MCML サーバースystemで通信する際に使用するポート番号を指定する。
戻り値	0 : 正常終了 1 : 指定した引数が適切でない(start, status, stop のいずれも指定されていない) 2 : 指定した引数が適切でない(start, status, stop 以外を指定) 3 : WFSTDM の実行に失敗	
説明	WFSTDM を起動する。	
使用例	# スタンドアロンモードで起動 \$ wfstdm start --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking --language=ja --standalone # MCML サーバースystemモードで起動 \$ wfstdm start --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking --language=ja --port=60001	

外部仕様書	プログラム名	WFSTDM	page	
			13	
	ページタイトル		Doc-No	

備考	スタンドアロンモードで実行する場合は事前にそのモードで WFSTDM をビルドする必要がある。 MCML サーバシステムモードにおいても同じ。
----	---


2-7. WFSTDM 停止

書式	wfstdm stop [--project=project]	
オプション	--project=project	処理対象となる WFSTDM のプロジェクトを指定する。省略した場合はカレントディレクトリをプロジェクトとする。
戻り値	0 : 正常終了 1 : 指定した引数が適切でない(start, status, stop のいずれも指定されていない) 2 : 指定した引数が適切でない(start, status, stop 以外を指定) 3 : 指定したプロジェクトが適切でない、または存在しない	
説明	WFSTDM を停止する。	
使用例	# WFSTDM 停止 \$ wfstdm stop --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking	
備考		

外部仕様書	プログラム名	WFSTDM	page	
	ページタイトル		14	
			Doc-No	

2-8. WFSTDM 状態出力


書式	wfstdm status [--project=project]	
オプション	--project=project	処理対象となる WFSTDM のプロジェクトを指定する。省略した場合はカレントディレクトリをプロジェクトとする。
戻り値	0 : 正常終了 1 : 指定した引数が適切でない(start, status, stop のいずれも指定されていない) 2 : 指定した引数が適切でない(start, status, stop 以外を指定)	
説明	WFSTDM の実行・停止状態を出力する。	
使用例	# WFSTDM 状態出力 \$ wfstdm status --project=/export/MCML/dmplatform/dmbuilder/user/wfstdm/cooking	
備考		

外部仕様書	プログラム名	WFSTDM	page	
			15	
	ページタイトル		Doc-No	

3. インタープリター

3-1. DB 検索

書式	DB_QUERY()	
オプション	-c	フレーズのクロージャーループを作成する。
	-f	concept による出力タグを発話の先頭に出力する。
	-h	ヘルプ(コマンドの使い方)を出力する。
	-l morph.fst	形態素解析時に使用するモデル(morph.fst)を指定する。
	-t lang_code	言語コード(ISO 639-1)を指定する。
引数	slu.xml	言語理解 XML を指定する。
	slu.fst	出力する言語理解 FST を指定する。
戻り値	0 : 正常終了 1 : 出力する言語理解 FST が指定されていない	
説明	言語理解 XML から FST を作成する。	
使用例	<pre>\$ make_slufst -l/export/MCML/dmbuilder/wfstdm/morph/CookingMorph.bin.fst slu/cooking.xml cooking_slu.bin.fst # 形態素解析モデルとして CookingMorph.bin.fst を指定 # 言語理解 XML として cooking.xml を指定 # 言語理解 FST の出力先として cooking_slu.bin.fst を指定</pre>	
備考		

外部仕様書	プログラム名	WFSTDM	page	
			16	
	ページタイトル		Doc-No	

4. DB 制御

4-1. WFSTDM で使用するテーブルの概念

WFSTDM で使用する対話パラメータは全て対話パラメータテーブル(t_dialog_param)で管理している。このテーブルからプロジェクトやその用途に応じて論理的なテーブルを作成するため、そのスキーマを定義した対話パラメータスキーマ定義テーブル(t_dialog_schema)がある。

- (1) 対話パラメータスキーマ定義テーブル 対話パラメータテーブルから作成する論理テーブルのカラム名と使用・未使用を定義したテーブル。
- (2) 対話パラメータテーブル ユーザーID、プロジェクト名、テーブル名をキーとして論理テーブルを作成し、対話パラメータを管理する。対話パラメータを管理する属性の名称は対話パラメータスキーマ定義テーブルで定義される。