

リアルタイム浸水予測システム
＜データ配信機能＞

利用の手引き

平成 31 年 3 月

国 土 技 術 政 策 総 合 研 究 所

目 次

1. 概要.....	1-1
2. 利用開始までの流れ.....	2-1
3. ダウンロード方法.....	3-3
4. データ仕様.....	4-6
4.1. CSV 形式のデータ仕様	4-6
4.1.1. 浸水深.....	4-6
4.1.2. 河川水位.....	4-8
4.1.3. 人孔内水位.....	4-10
4.1.4. 下水道管内観測水位	4-11
4.2. タイル形式のデータ仕様	4-12

1. 概要

リアルタイム浸水予測システムでは、リアルタイムデータの配信サービスを行っております。申込後、ユーザーID およびパスワードが発行された方は表 1-1 に示す 5 種類のデータをダウンロードすることができます。

表 1-1 配信データの種類

データ種類	データ形式	予測時間	更新頻度	配信方法
河川水位	CSV	60 分先	10 分	FTP
人孔水位	CSV	60 分先	10 分	FTP
下水管内観測水位	CSV	(観測値データ)	10 分	FTP
浸水深	CSV	60 分先	10 分	FTP
	タイル	60 分先	10 分	WebAPI (https)

2. 利用開始までの流れ

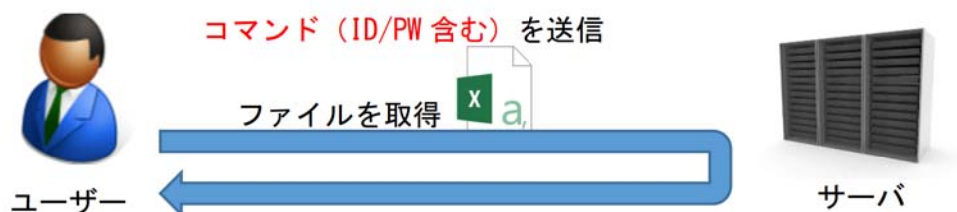
手順①：システム TOP ページから「データ配信申し込み書」をダウンロードしてください。



手順②：ユーザーは国総研に次頁のデータ配信申し込み書をメール等で提出してください。

手順③：国総研がユーザーID/PW を発行し、利用の手引きを配布します。

<データ取得のイメージ>



データ配信機能申込書

以下を記入し送付先メールアドレスまで「データ配信機能申込書」をメールで送付してください。

	記入欄
所属（必須）	
氏名	
メールアドレス（必須）	

送付先メールアドレス

sip_faq@ss.pacific.co.jp

国土交通省 国土技術政策総合研究所 河川研究部 水害研究室
メールアドレス：nil-suigai@mlit.go.jp
対応時間：平日（月～金）9:00～17:00 土日祝日を除く

3. ダウンロード方法

CSV 形式のデータ（浸水深、河川水位、人孔水位、下水管内観測水位）は所定の FTP サーバに最新のデータが格納される仕組みとしています。ユーザーID およびパスワードを入力して、ダウンロードしてください。

タイル形式の浸水深データは所定の URL にアクセスすると WebAPI が最新の基準時刻におけるダウンロード可能な浸水深タイルの URL リストを JSON 形式で配信する仕組みとしています。これを基に浸水深タイルデータをダウンロードすることも可能ですし、利用者側構築したアプリケーションに組み込んで自動的に最新の基準時間におけるタイル画像にアクセスすることが可能となります。

図 3-1 および図 3-2 にデータ配信のイメージを示します。

※システムメンテナンス作業などの理由により、一時的にデータダウンロードができなくなる場合がありますので、データ取得に失敗した場合は繰り返し、取得処理することを推奨いたします。

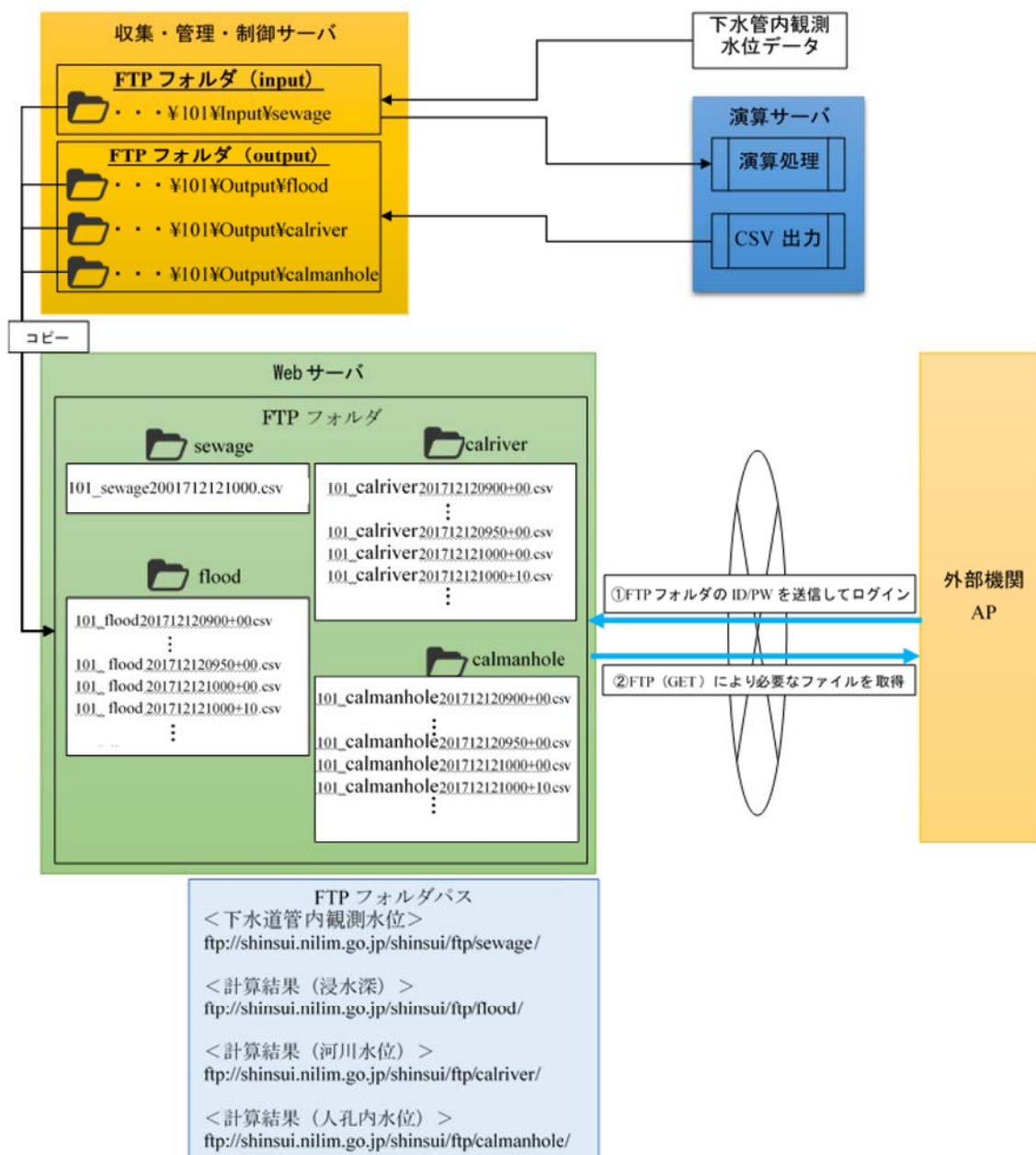


図 3-1 FTP 方式による CSV データの配信イメージ

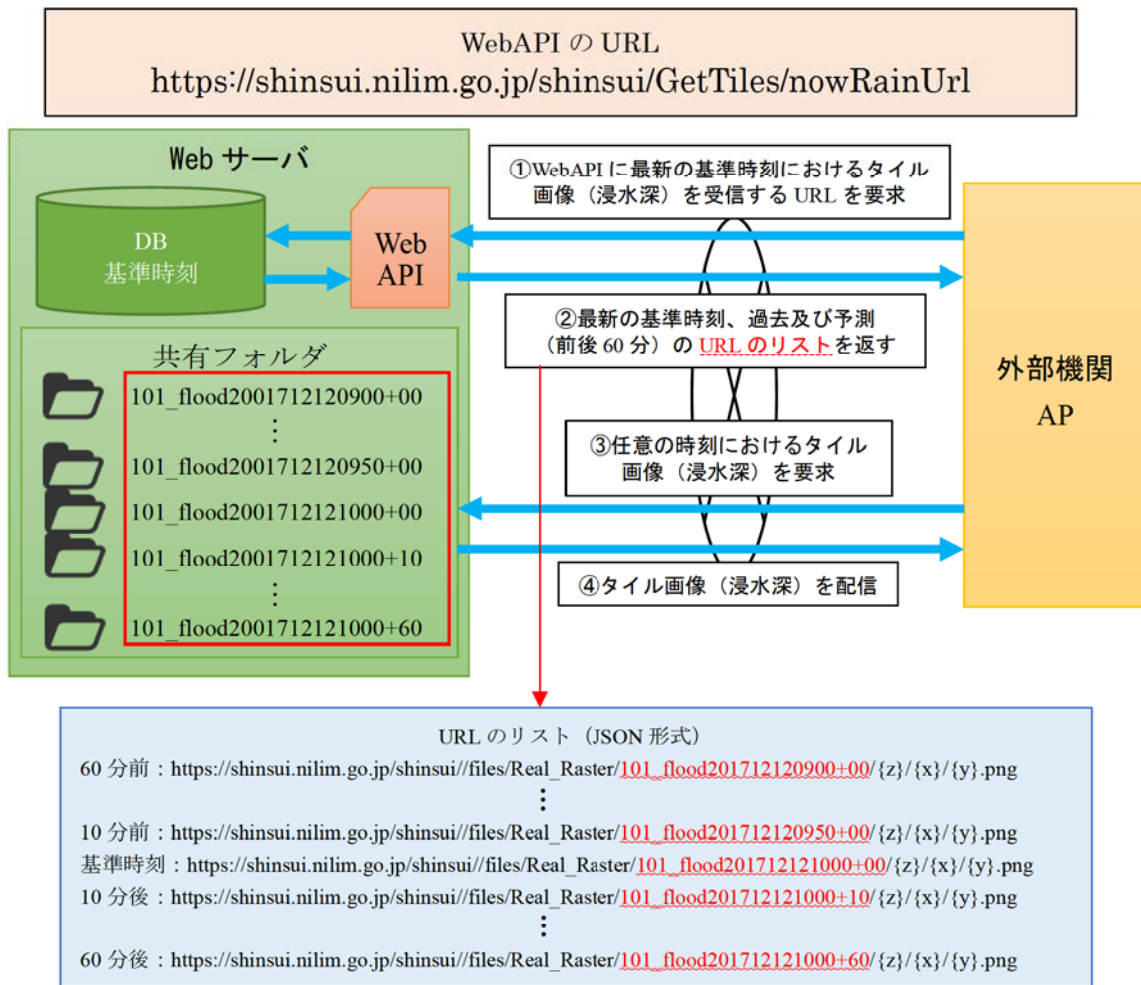


図 3-2 WebAPI 方式による浸水深タイルデータの配信イメージ

4. データ仕様

4.1. CSV 形式のデータ仕様

浸水深、河川水位、人孔内水位及び下水管内観測水位のデータ仕様は以下のようになっております。

計算結果については基準時刻（予測の初期値としているデータの時刻）の前後 1 時間のデータ、下水管内観測水位については基準時刻のデータを配信するものとしております。

4.1.1. 浸水深

表 4-1 浸水深メッシュのデータ仕様

項目	内容	
ファイル単位	過去	基準時間から 60 分前まで（10 分間隔）
	現況	基準時間（10 分間隔）
	予測	基準時間から 60 分後まで（10 分先、20 分先、30 分先、40 分先、50 分先、60 分先）
ファイル形式	フラットファイル形式	.csv（UTF-8）
データ単位	浸水深（m）	m（小数点 2 桁）※10cm 未満は出力しない。
保存先ディレクトリ	ftp://shinsui.nilim.go.jp/shinsui/ftp/（モデル ID） /flood/	

※モデル ID は、神田川モデル=101

- ファイル命名規則

浸水深メッシュデータのファイル命名規則は以下のとおりです。

浸水深ファイル（現況）

***_floodYYYYMMDDHHMM+00.csv	
T	—————
モデル ID（※）	予測基準時間（年月日時分）

※モデル ID は、神田川モデル=101

例) 101_flood201412232200+00.csv

浸水深ファイル (予測)

***_floodYYYYMMDDHHMM+FF.csv

モデル ID (※) 予測基準時間 (年月日時分) 予測時間 (+10 の場合 10 分後)

※モデル ID は、神田川モデル=101

例) 101_flood201412232200+10.csv

- ファイル構造

浸水深メッシュ（現況・予測）データのファイル構造は以下のとおりです。

101 ←モデル ID (※)

2014,12,25,03,20 ←現況時間 (予測の場合は予測に用いた現況の時間) yyyy,mm,dd,hh,mm

0 ←予測時間 (現況の場合は 0) 0~60

5339356721120,1.42

5339356721122,2.01

5339356721124,1.14

5339356721154,1.10

5339356721156,1.37

5339356721157,1.20

5339356721158,0.72

↑

細分メッシュ ID (13 桁),浸水深 (m)

※浸水深のあるメッシュのみ出力する。

※10cm 未満は出力しない。

※モデル ID は、神田川モデル=101

4.1.2. 河川水位

表 4-2 河川水位のデータ仕様

項目	内容	
ファイル単位	過去	基準時間から 60 分前まで（10 分間隔）
	現況	基準時間（10 分間隔）
	予測	基準時間から 60 分後まで（10 分先、20 分先、30 分先、40 分先、50 分先、60 分先）
ファイル形式	フラットファイル形式	.csv（UTF-8）
データ単位	水位（T.P.m）	T.P.m（小数点 2 桁）
保存先ディレクトリ	ftp://shinsui.nilim.go.jp/shinsui/ftp/ （モデル ID）/calriver	

※モデル ID は、神田川モデル=101

- ファイル命名規則

計算水位（河川）データのファイル命名規則は以下のとおりです。

計算水位（河川）ファイル（現況）

***_calriverYYYYMMDDHHMM+00.csv	
T	T
モデル ID（※）	予測基準時間（年月日時分）

※モデル ID は、神田川モデル=101

例) 101_calriver201412232200+00.csv

計算水位（河川） ファイル（予測）

***_calriverYYYYMMDDHHMM+FF.csv

モデル ID (※) 予測基準時間 (年月日時分) 予測時間 (+10 の場合 10 分後)

※モデル ID は、神田川モデル=101

計算水位（河川）データのファイル構造は以下のとおりです。

101 ←モデル ID (※)
 2014,12, 25,03,20 ←現況時間 (予測の場合は予測に用いた現況の時間) yyyy,mm,dd,hh,mm
 0 ←予測時間 (現況の場合は 0) 0~60
 1011000001,1.42
 1011000002,2.01
 1011000007,0.72
 ↑
 地点 ID (10 桁), 計算水位 (T.P.m)

※モデル ID は、神田川モデル=101

4.1.3. 人孔内水位

表 4-3 人孔内水位のデータ仕様

項目	内容	
ファイル単位	過去	基準時間から 60 分前まで（10 分間隔）
	現況	基準時間（10 分間隔）
	予測	基準時間から 60 分後まで（10 分先、20 分先、30 分先、40 分先、50 分先、60 分先）
ファイル形式	.csv（UTF-8）	
データ単位	水位（T.P.）m（小数点 2 桁）	
保存先ディレクトリ	ftp://shinsui.nilim.go.jp/shinsui/ftp/ （モデル ID） /calmanhole	

人孔内水位データのファイル命名規則は以下のとおりです。

人孔内水位 (現況)

*** calmanhole **YYYYMMDDHHMM+00.csv**

モデル ID (※)

予測基準時間 (年月日時分)

※モデル ID は、神田川モデル=101

例) 101 calmanhole201412232200+00.csv

人孔内水位 (予測)

*** calmanhole YYYYMMDDHHMM+FF.csv

モデル ID (※)

予測基準時間 (年月日時分)

予測時間 (+10 の場合 10 分後)

※モデル ID は、神田川モデル=101

例) 101_calmanhole201412232200+00.csv

人孔内水位データのファイル構造は以下のとおりです。

101	←モデル ID (※)	}	ヘッダー 情報
2014,12,25,03,20	←現況時間 (予測の場合は予測に用いた現況の時間) yyyy,mm,dd,hh,mm		
0	←予測時間 (現況の場合は 0) 0~60		
01001,1.42	←識別情報 (5 桁),人孔内水位 (T.P.m)		
01002,2.01	←識別情報 (5 桁),人孔内水位 (T.P.m)	}	(T.P.m 単位) 人 孔 内 水 位

※モデル ID は、神田川モデル=101

4.1.4. 下水道管内観測水位

表 4-4 下水道管内観測水位のデータ仕様

項目	内容
ファイル単位	観測時刻 (10 分間隔)
ファイル形式	.csv (UTF-8)
データ単位	m (小数点 2 桁)
保存先ディレクトリ	ftp://shinsui.nilim.go.jp/shinsui/ftp/ (モデル ID) /sewage

※モデル ID は、神田川モデル=101

下水管内水位データのファイル命名規則は以下のとおりです。

***	_sewage	YYYYMMDDHHMM	.csv
↓		↓	
モデル ID (※)		観測時刻	

※モデル ID は、神田川モデル=101

例) 101_sewage201412242300.csv

リアルタイムデータベースから配信される下水管内水位データのファイル構造は以下のとおりです。

101	←モデル ID (※)	}	ヘッダー 情報 内 水 管
00000,00000,00000,...	←識別情報 (5 桁)		
00,00,00,...	←水位計名		
201412241420,0.00,0.00,0.00,...	←観測日時 (=yyyymmddhhmm),水位 (=m)		

※モデル ID は、神田川モデル=101

4.2. タイル形式のデータ仕様

配信するデータは、リアルタイム浸水予測システムでリアルタイムに生成される計算結果（浸水深）のタイル画像です。

データ配信する計算結果（浸水深）のタイル画像のデータ仕様について以下に示します。国土地理院のタイル仕様に則った仕様としています。

表 4.2-1 浸水深メッシュのタイル画像のデータ仕様

項目	仕様
データ項目	浸水深（神田川浸水予測モデル）
タイル画像作成時刻（10 分単位）	基準時間（観測時間から 20 分後） 基準時間から 60 分前まで（過去実績） 基準時間から 60 分後まで（予測）
データ形式	PNG 形式
座標情報	地理院タイル仕様
座標系	移動経度（WGS84）
ファイル命名規則	地理院タイル仕様 {ID}/{z}/{x}/{y}.png {ID}：データ ID {x}：タイル座標の X 値 {y}：タイル座標の Y 値 {z}：ズームレベル データ ID は以下である。 【浸水深メッシュ】 101_floodYYYYMMDDHHMM+FF ※FF：00～60
タイル画像 1 枚の大きさ	縦 256 ピクセル、横 256 ピクセル
浸水深=0 の表現方法	透過色
作成ズームレベル	{z}=12,13,14,15,16

リアルタイム浸水予測システム システム操作マニュアル（管理者向け）

平成 31 年 3 月

国土技術政策総合研究所
パシフィックコンサルタンツ株式会社

改訂履歴

改訂年月日	改訂内容／理由
2016/3/22	初版作成
2018/3/19	C・X 合成レーダ雨量データおよび人孔水位受信に伴う改訂
2018/3/4	テレメータ雨量削除に伴う改訂
2019/3/15	浸水常襲箇所レイヤーの設定変更方法追加に伴う改訂

リアルタイム浸水予測システム
システム操作マニュアル（管理者向け）

目次

1. 本マニュアルの対象.....	1-1
2. リモートデスクトップの設定.....	2-1
2.1 VPN クライアントの設定と接続.....	2-1
2.2 リモートデスクトップの接続	2-6
2.3 リモートデスクトップの切断	2-8
3. 擬似リアルタイム浸水システム（模擬）	3-1
3.1 システムの起動.....	3-1
3.2 メインメニュー.....	3-1
3.3 データの初期化.....	3-2
3.4 仮想日時設定	3-3
3.5 訓練モードの終了	3-3
4. リアルタイム浸水システム	4-4
4.1 システムの起動.....	4-4
4.2 メインメニュー.....	4-4
4.3 リアルタイム処理の開始	4-5
5. リアルタイム処理制御システム（疑似リアルタイム制御システム）	5-1
5.1 起動	5-1
5.2 TOP 画面.....	5-2
5.3 リアルタイムデータ収集格納処理	5-3
5.4 合成雨量生成処理	5-4
5.5 リアルタイムデータ配信処理	5-5
5.6 演算モデルデータ取得処理.....	5-6
5.7 演算処理エラー一覧.....	5-7
5.8 配信用空間データベース格納処理	5-8
5.9 タイル化処理	5-10
6. 浸水常襲箇所レイヤーの追加	6-11

1. 本マニュアルの対象

本マニュアルは、都市中小河川におけるリアルタイム浸水情報配信システムの収集管理制御機能に関する操作マニュアルであり、疑似リアルタイム処理およびリアルタイム処理の操作について説明するものです。

※クラウドサーバへのアクセス手順およびユーザ名、パスワード等が記載されています。取り扱いには十分注意してください。

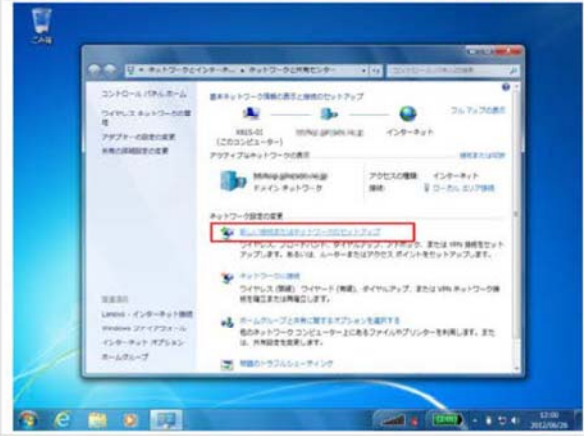
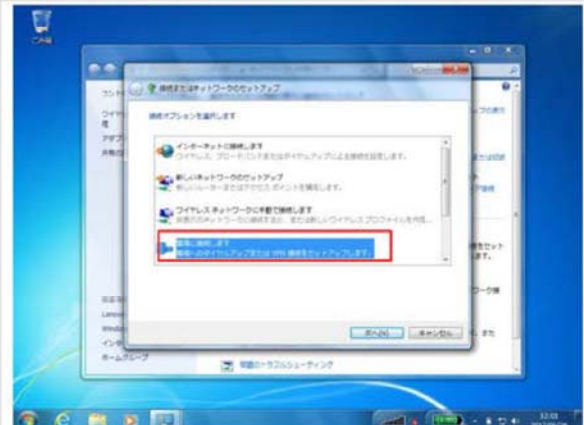
2. リモートデスクトップの設定

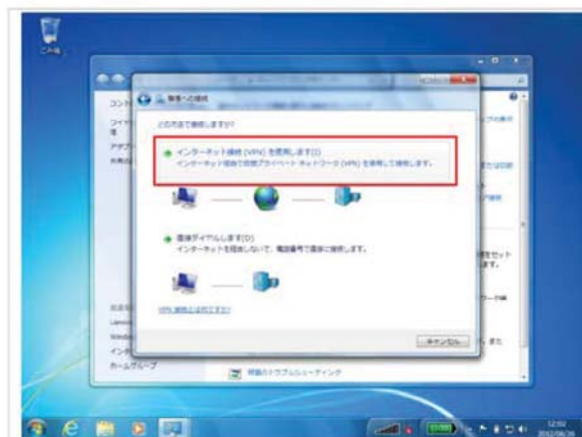
収集管理制御システムは、クラウド上の仮想サーバにリモート接続することで操作を行うことができます。

このため、以下の方法で、VPN クライアントの設定を行い、VPN でクラウドのネットワークワークに接続、リモートデスクトップで仮想サーバに接続する必要があります。

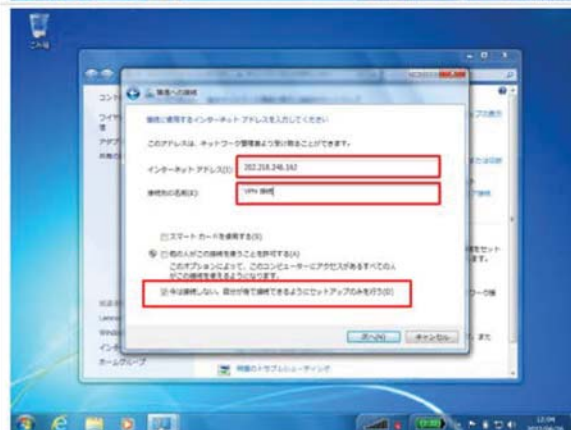
以下にその手順を示します。

2.1 VPN クライアントの設定と接続

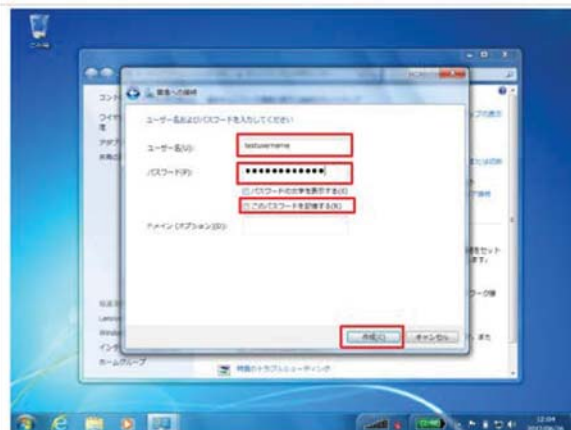
	タスクトレイからネットワークのアイコンをクリックし「ネットワークと共有センターを開く」をクリックします。
	2 「ネットワーク設定の変更」にある「新しい接続またはネットワークのセットアップ」をクリックします。
	3 「接続オプションを選択します」で「職場に接続します」を選択し「次へ」ボタンを押します。



- 4 [どの方法で接続しますか?] で
[インターネット接続(VPN)を使用します] をクリックします。

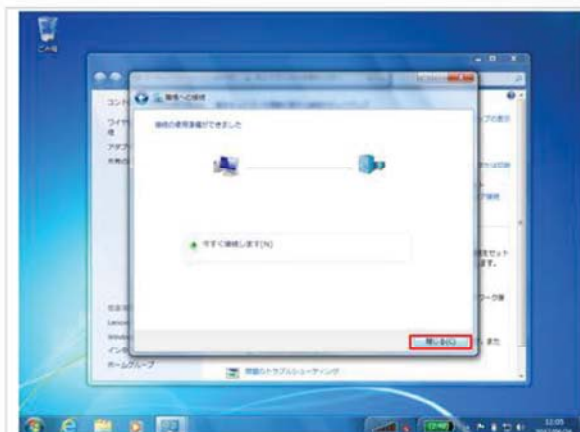


- 5 [インターネットアドレス] にセルフポータルのグローバル IP アドレス (ソース) を入力します。
- 6 [接続先の名前] にこの設定の名前を入力します。この値は実際に接続しようとする際に表示される名前になります。
- 7 [今は接続しない。自分が後で接続できるようにセットアップのみを行う] をチェックします。
- 8 [次へ] ボタンを押します。

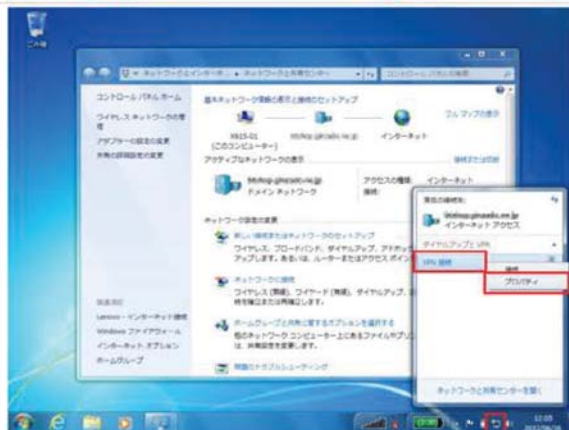


- 9 [ユーザー名] にセルフポータルで設定した『ユーザー名』を入力します。
- 10 [パスワード] にセルフポータルで設定した『パスワード』を入力します。パスワードを保存するときは [このパスワードを記憶する] をチェックします。
- 11 [作成] ボタンを押します。

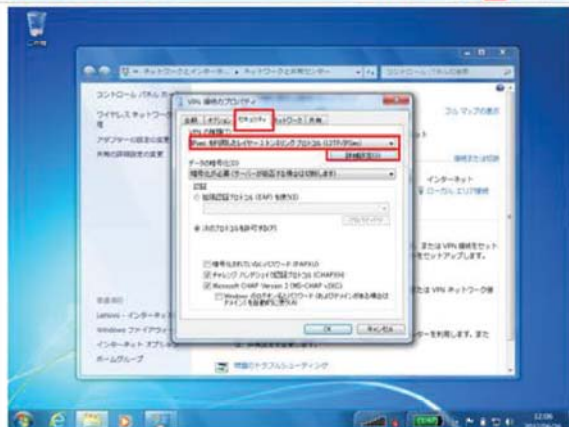
手順5のIPアドレスは [REDACTED]
 手順9のユーザ名は [REDACTED]
 手順10のパスワードは [REDACTED]



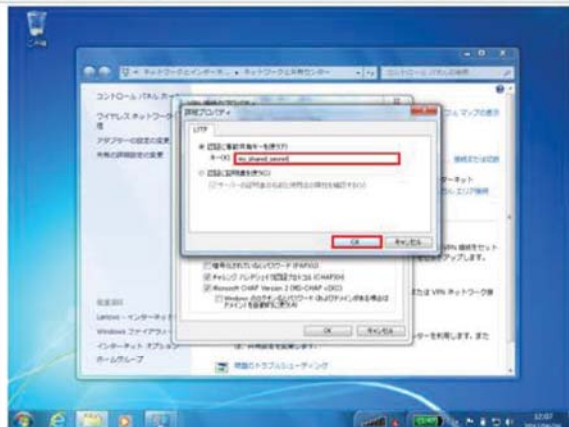
12 「閉じる」ボタンを押します。



13 タスクトレイのネットワークのアイコンをクリックし、作成された設定を右クリックして「プロパティ」を選択します。



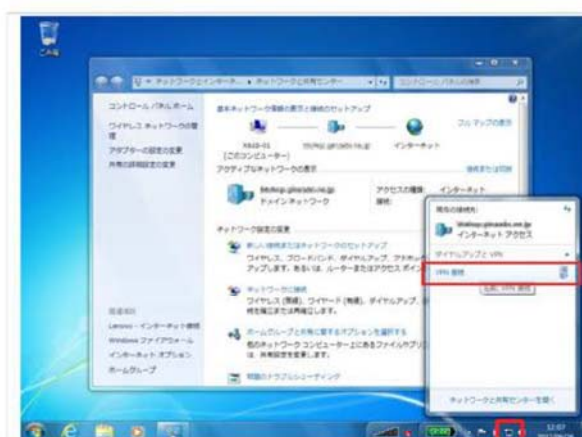
14 「セキュリティ」タブを開き「VPNの種類」で「IPsec を利用したレイヤー 2 トンネリングプロトコル (L2TP/IPsec)」を選択し「詳細設定」ボタンを押します。



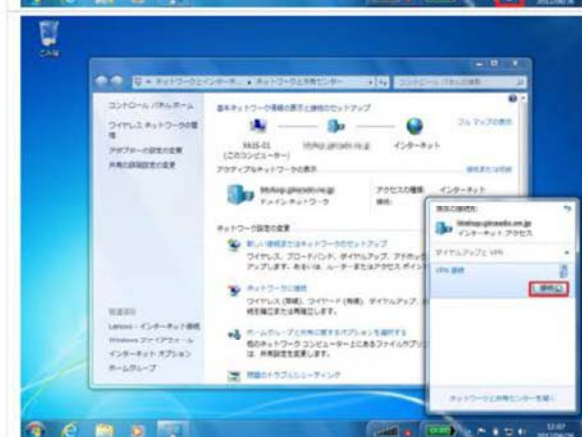
15 「L2TP」タブで「認証に事前共有キーを使う」を選択しセルフポータルで表示された『IPSec プリシェアードキー』を入力し「OK」ボタンを押します。

手順 15 の IPsec プリシェアードキーは





16 タスクトレイからネットワークのアイコンをクリックし[ダイアログアップと VPN]の中から接続する設定をクリックします。



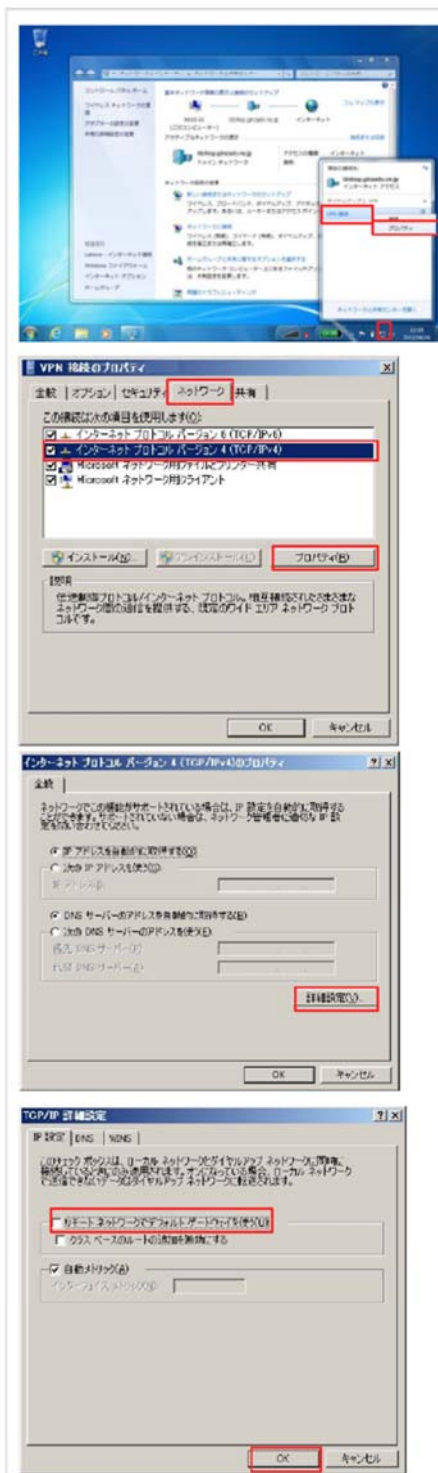
17 選択した設定の[接続] ボタンを押します。



18 [ユーザー名]と[パスワード]を確認し[接続] ボタンを押します。

19 問題なく接続されれば左の図のように[アクティブなネットワークの表示]に上記で作成した設定が表示されます。

20 切断するときはタスクトレイからネットワークのアイコンをクリックし接続中の設定の[切断] ボタンを押します。



21 【注意補足】

WindowsOS では、リモートアクセス VPNを行うと全ての通信がVPN側に向かい、その他の通信が切断されてしまいます。

これを回避するには、以下の設定を行ってください。

- ◆ タスクバーのネットワークアイコンをクリック→VPN設定を右クリック→[プロパティ] をクリックします。
- ◆ [ネットワーク] タブ→[インターネットプロトコルバージョン4(TCP/IPv4)] を選択して[プロパティ] をクリック。
- ◆ [詳細設定] をクリック。
- ◆ [リモートネットワークでデフォルトゲートウェイを使う]、のチェックボックスを外す。
- ◆ 開いているウィンドウを [OK] で閉じます。

2.2 リモートデスクトップの接続

VPN 接続が完了した状態で、仮想サーバにリモート接続します。

[Windows メニュー] → [すべてのプログラム] → [アクセサリ] →
[リモートデスクトップ接続] を選択します。

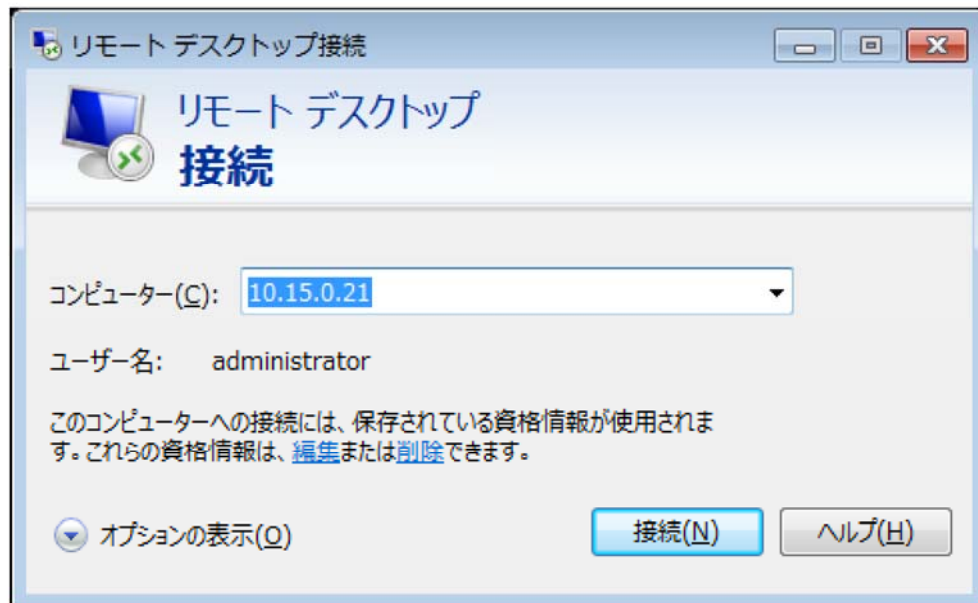


図 2-1 リモートデスクトップ接続画面

コンピューター (C) に「10.15.0.21」と入力
[接続] をクリック

下図のウィンドウが表示された場合には、[接続] をクリック

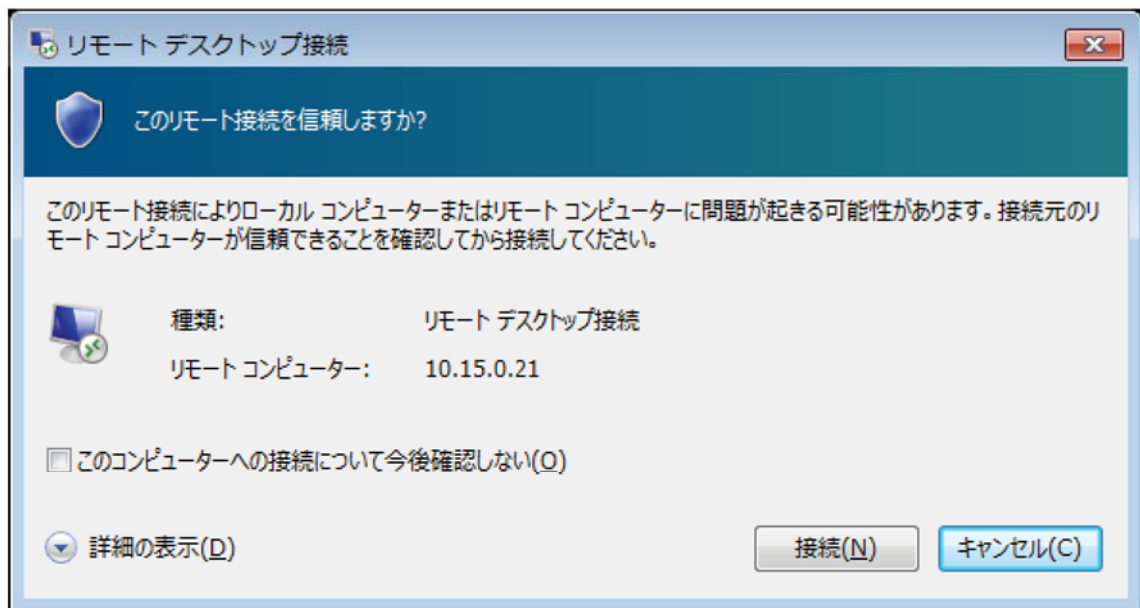


図 2-2 リモートデスクトップ接続画面その 2

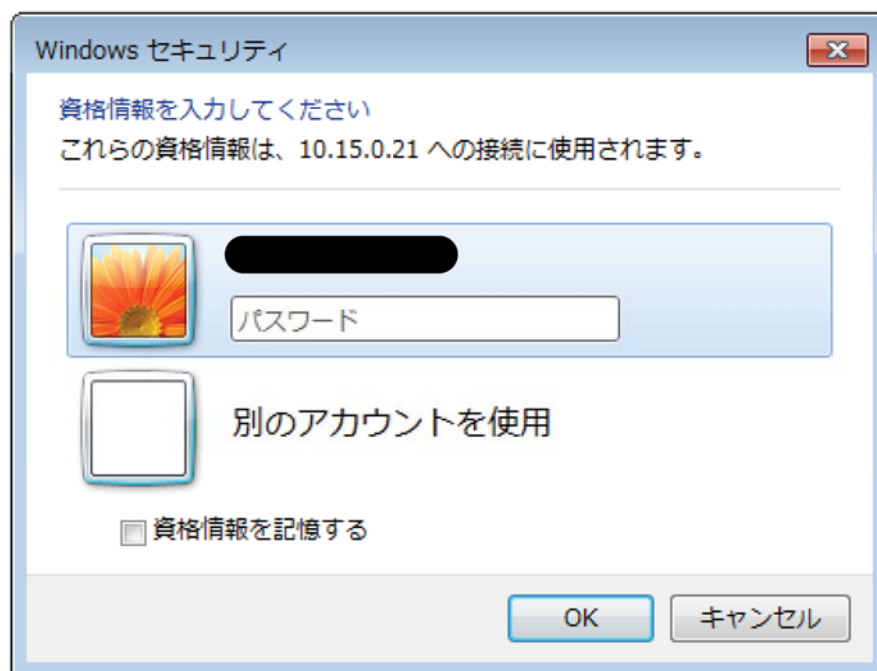


図 2-3 リモートデスクトップ接続ログイン画面

ユーザ名 : [Redacted]
パスワード : [Redacted]
を入力し、[OK] をクリック

2.3 リモートデスクトップの切断

リモートデスクトップ接続を切断する場合には、画面上部の接続バー右端の [×] 印をクリックします。



図 2-4 接続バー

下図の確認ウィンドウが表示されますので、[OK] をクリックします。

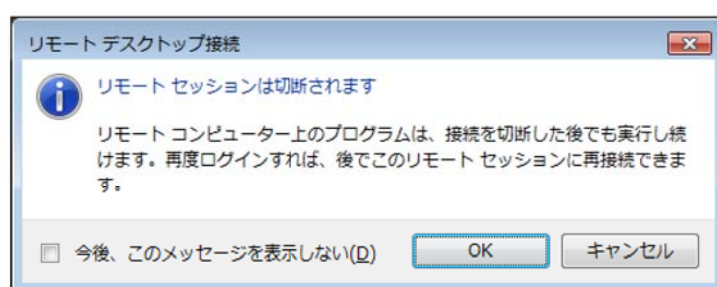


図 2-5 切断確認ウィンドウ

※仮想サーバは、常時起動状態ですので、「ログオフ」や「電源切断」は絶対に行わないでください。

3. 疑似リアルタイム浸水システム（模擬）

ここでは疑似リアルタイム浸水システム（模擬）を用いて、訓練モードを起動する方法を説明します。

3.1 システムの起動



図 3-1 リアルタイム浸水システム ショートカットアイコン

ショートカットアイコンをダブルクリックすることで起動します。

3.2 メインメニュー

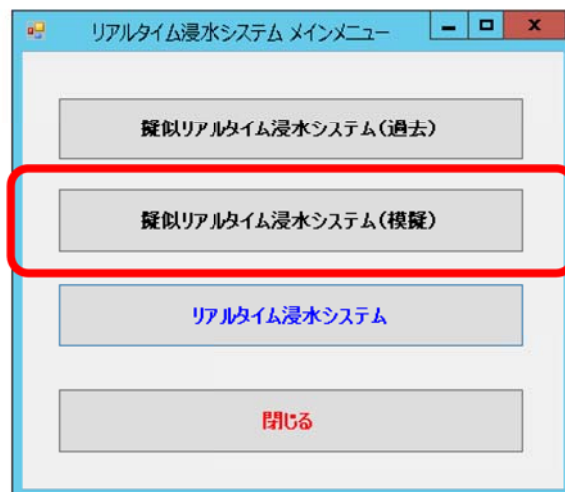


図 3-2 リアルタイム浸水システム メインメニュー

[疑似リアルタイム浸水システム（模擬）] をクリックします。

3.3 データの初期化

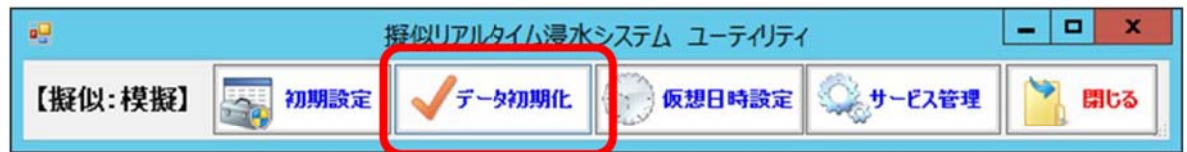
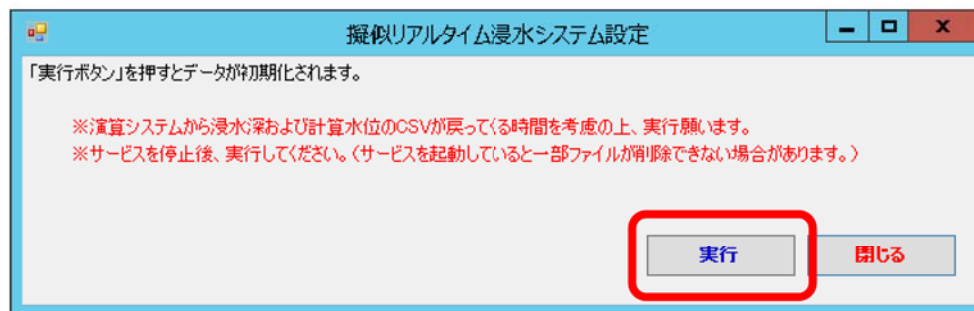


図 3-3 ユーティリティ

① 「データ初期化」をクリックします。



② ¥¥10.15.0.220¥Test_Raster フォルダの中身を空にします。

3.4 仮想日時設定

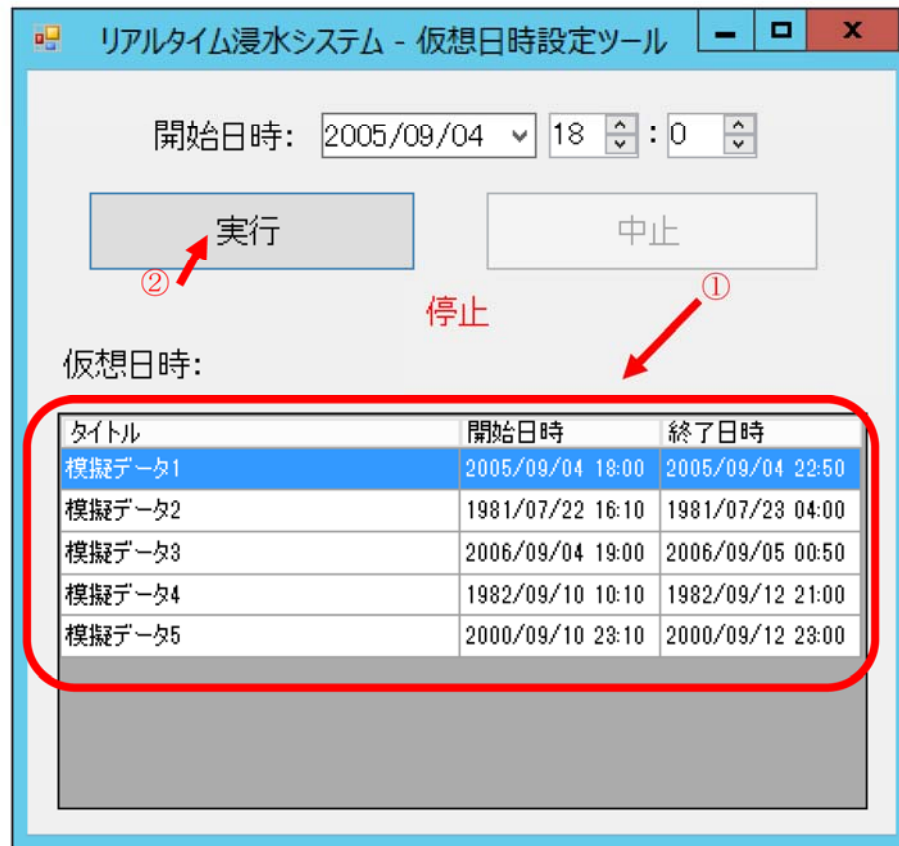


図 3-4 仮想日時設定画面

- ① 下欄からデータを選択
- ② 「実行」をクリック

事前にセットされたデータがシステムに登録され、訓練モードとなります。

3.5 訓練モードの終了

「中止」をクリックすると訓練モードは終了します。

4. リアルタイム浸水システム

ここではリアルタイム浸水システムを用いて、リアルタイムモードを起動する方法を説明します。

4.1 システムの起動



図 4-1 リアルタイム浸水システム ショートカットアイコン

ショートカットアイコンをダブルクリックすることで起動します。

4.2 メインメニュー

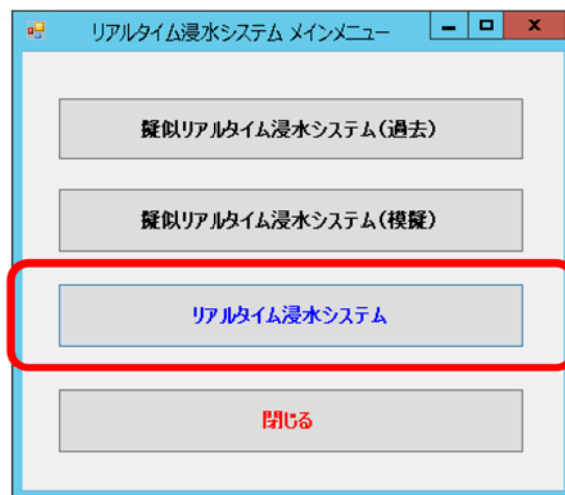


図 4-2 リアルタイム浸水システム メインメニュー

「リアルタイム浸水システム」をクリックします。

4.3 リアルタイム処理の開始

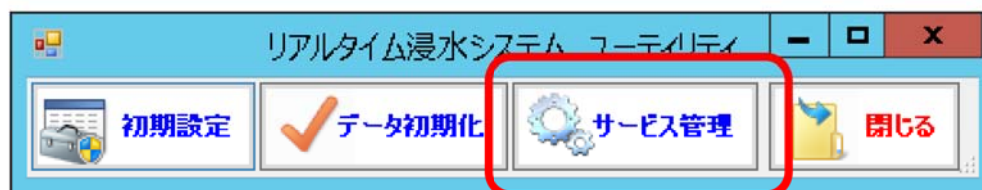


図 4-3 ユーティリティ

- ①[サービス管理]をクリックします。
- ②サービス管理の[一括開始]をクリックします。



図 4-4 サービス管理

5. リアルタイム処理制御システム（疑似リアルタイム制御システム）

5.1 起動

Internet Explorer を起動し、以下の URL を入力するか、デスクトップ上のアイコンをクリックします。

リアルタイム処理の制御システムの URL は

<http://localhost/rtf/index.php>

疑似リアルタイム制御システムの URL は

<http://localhost/vtf/index.php>

です。

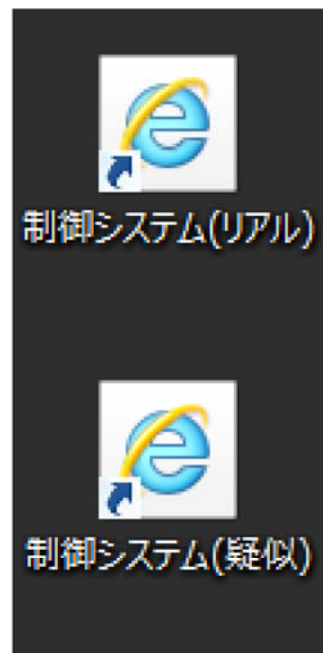


図 5-1 制御システムショートカットアイコン

5.2 TOP 画面

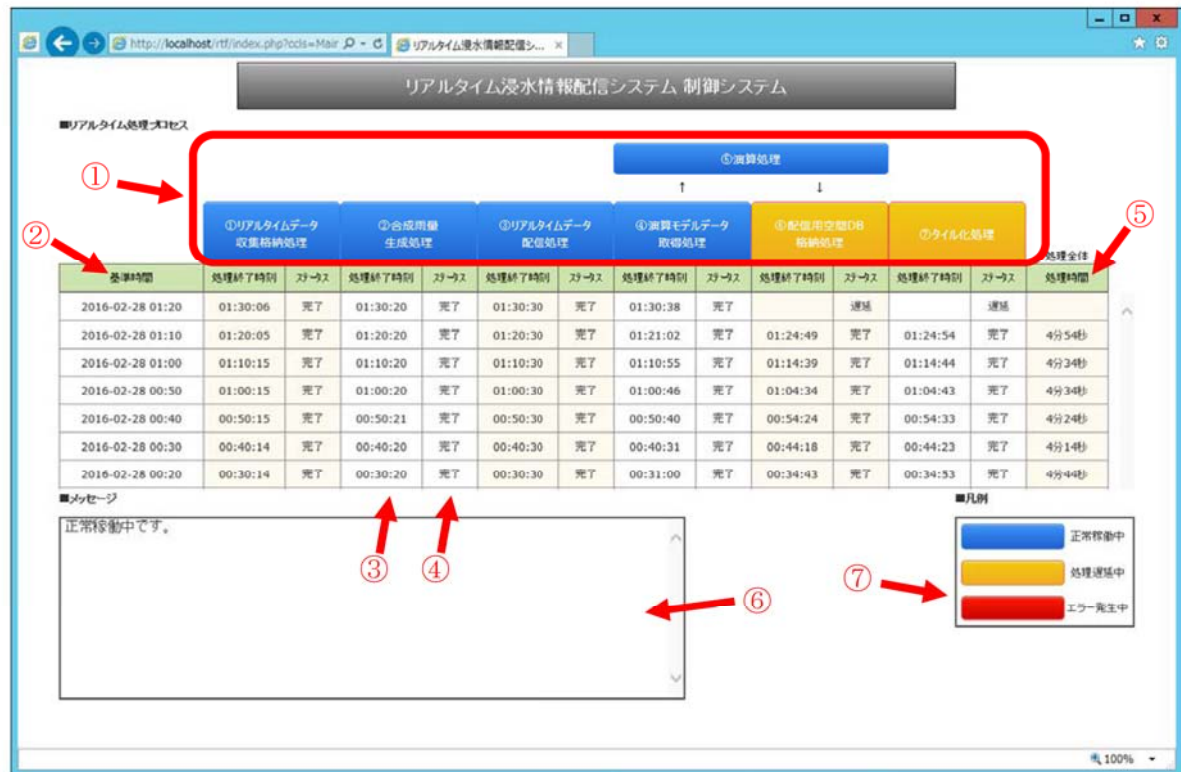


図 5-2 制御システム TOP 画面

- ①各処理の詳細表示画面への遷移ボタン
クリックすることで遷移
- ②処理開始時刻を表示
- ③各処理の終了時刻を表示
- ④処理ステータスを表示
- ⑤全体処理時間を表示
- ⑥稼働状況表示を表示
- ⑦稼働状況色別表示の凡例を表示

5.3 リアルタイムデータ収集格納処理

リアルタイム洪水情報配信システム 制御システム

① テレメータ水位(河川)収集格納

観測時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:00	0.6秒	完了
2016-02-28 01:10	01:20:00	0.6秒	完了
2016-02-28 01:00	01:10:10	0.8秒	完了
2016-02-28 00:50	01:00:10	0.6秒	完了
2016-02-28 00:40	00:50:10	0.5秒	完了
2016-02-28 00:30	00:40:09	0.5秒	完了

② C-X合成レーダ雨量収集格納

基準時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:04	1秒	完了
2016-02-28 01:19	01:30:03	0.9秒	完了
2016-02-28 01:18	01:30:03	1.1秒	完了
2016-02-28 01:17	01:30:01	0.8秒	完了
2016-02-28 01:16	01:30:01	0.9秒	完了
2016-02-28 01:15	01:25:04	0.9秒	完了

③ 高解像度降水ナウキャスト収集格納

基準時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:01	1.1秒	完了
2016-02-28 01:15	01:25:01	1.5秒	完了
2016-02-28 01:10	01:20:01	1.2秒	完了
2016-02-28 01:05	01:15:11	1秒	完了
2016-02-28 01:00	01:10:10	1.1秒	完了

④ 下水管内観測水位収集格納

基準時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:06	5.7秒	完了
2016-02-28 01:15	01:25:05	5.8秒	完了
2016-02-28 01:10	01:20:05	5.5秒	完了
2016-02-28 01:05	01:15:15	5.6秒	完了
2016-02-28 01:00	01:10:15	5.6秒	完了

⑤ 稼働状況表示

⑥ 稼働状況色別表示の凡例

- 正常稼働中 (Blue)
- 処理進捗中 (Yellow)
- エラー発生中 (Red)

■メッセージ

正常稼働中です。

図 5-3 リアルタイムデータ収集格納処理 画面

①テレメータ水位（河川）収集格納

テレメータ水位（河川）の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②C-X 合成レーダ雨量収集格納

C-X 合成レーダ雨量情報の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

③高解像度降水ナウキャスト収集格納

高解像度降水ナウキャストの基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

④下水管内観測水位収集格納

下水管内観測水位の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑤稼働状況表示

⑥稼働状況色別表示の凡例

5.4 合成雨量生成処理

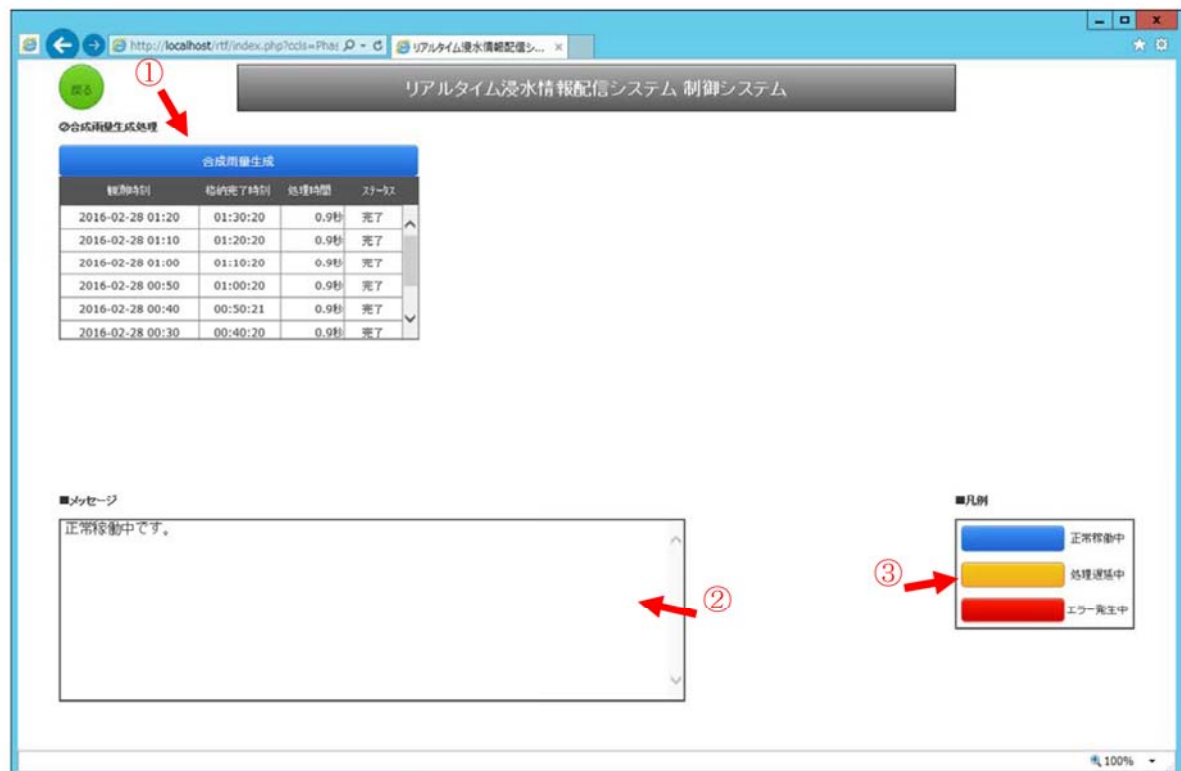


図 5-4 合成雨量生成処理 画面

①合成雨量生成

合成雨量生成の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②稼働状況表示

③稼働状況色別表示の凡例

5.5 リアルタイムデータ配信処理

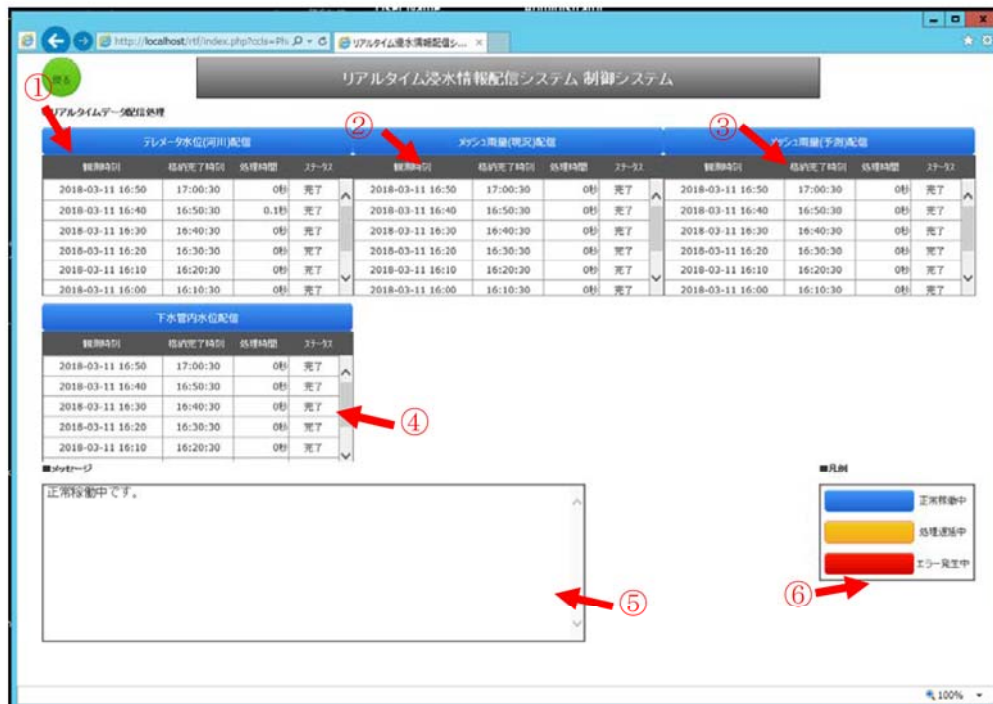


図 5-5 リアルタイムデータ配信処理 画面

①テレメータ水位（河川）配信

テレメータ水位（河川）の観測時刻、格納（配信）完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②メッシュ雨量（現況）配信

メッシュ雨量（現況）の観測時刻、格納（配信）完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

③メッシュ雨量（予測）配信

メッシュ雨量（予測）の観測時刻、格納（配信）完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

④下水管内水位配信

下水管内水位の観測時刻、格納（配信）完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑤稼働状況表示

⑥稼働状況色別表示の凡例

5.6 演算モデルデータ取得処理

リアルタイム浸水情報配信システム 制御システム

① モデルデータ取得処理

テレメータ水位(河川)取得				メッシュ雨量(現況)取得				メッシュ雨量(予測)取得			
観測時刻	格納完了時刻	処理時間	ステータス	観測時刻	格納完了時刻	処理時間	ステータス	観測時刻	格納完了時刻	処理時間	ステータス
2018-03-11 16:50	17:00:48	17.9秒	完了	2018-03-11 16:50	17:00:48	17.8秒	完了	2018-03-11 16:50	17:00:48	17.8秒	完了
2018-03-11 16:40	16:50:42	11.8秒	完了	2018-03-11 16:45	17:00:48	5分18秒	完了	2018-03-11 16:45	17:00:48	5分18秒	完了
2018-03-11 16:30	16:41:11	40.9秒	完了	2018-03-11 16:40	16:50:42	11.6秒	完了	2018-03-11 16:40	16:50:42	11.6秒	完了
2018-03-11 16:20	16:31:09	38.9秒	完了	2018-03-11 16:35	16:50:42	5分12秒	完了	2018-03-11 16:35	16:50:42	5分12秒	完了
2018-03-11 16:10	16:20:42	11.9秒	完了	2018-03-11 16:30	16:41:11	40.8秒	完了	2018-03-11 16:30	16:41:11	40.8秒	完了
2018-03-11 16:00	16:10:57	28.9秒	完了	2018-03-11 16:25	16:41:11	5分41秒	完了	2018-03-11 16:25	16:41:11	5分41秒	完了

下水管内水位取得			
観測時刻	格納完了時刻	処理時間	ステータス
2018-03-11 16:50	17:00:48	17.9秒	完了
2018-03-11 16:40	16:50:42	11.7秒	完了
2018-03-11 16:30	16:41:11	40.9秒	完了
2018-03-11 16:20	16:31:09	38.9秒	完了
2018-03-11 16:10	16:20:42	11.9秒	完了

メッセージ
正常稼働中です。

凡例
■ 正常稼働中
■ 処理遅延中
■ エラー発生中

図 5-6 演算モデルデータ取得処理 画面

①テレメータ水位（河川）取得

テレメータ水位（河川）の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②メッシュ雨量（現況）取得

メッシュ雨量（現況）の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

③メッシュ雨量（予測）取得

メッシュ雨量（予測）の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

④下水管内水位取得

下水管内水位の観測時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑤稼働状況表示

⑥稼働状況色別表示の凡例

5.7 演算処理エラー一覧

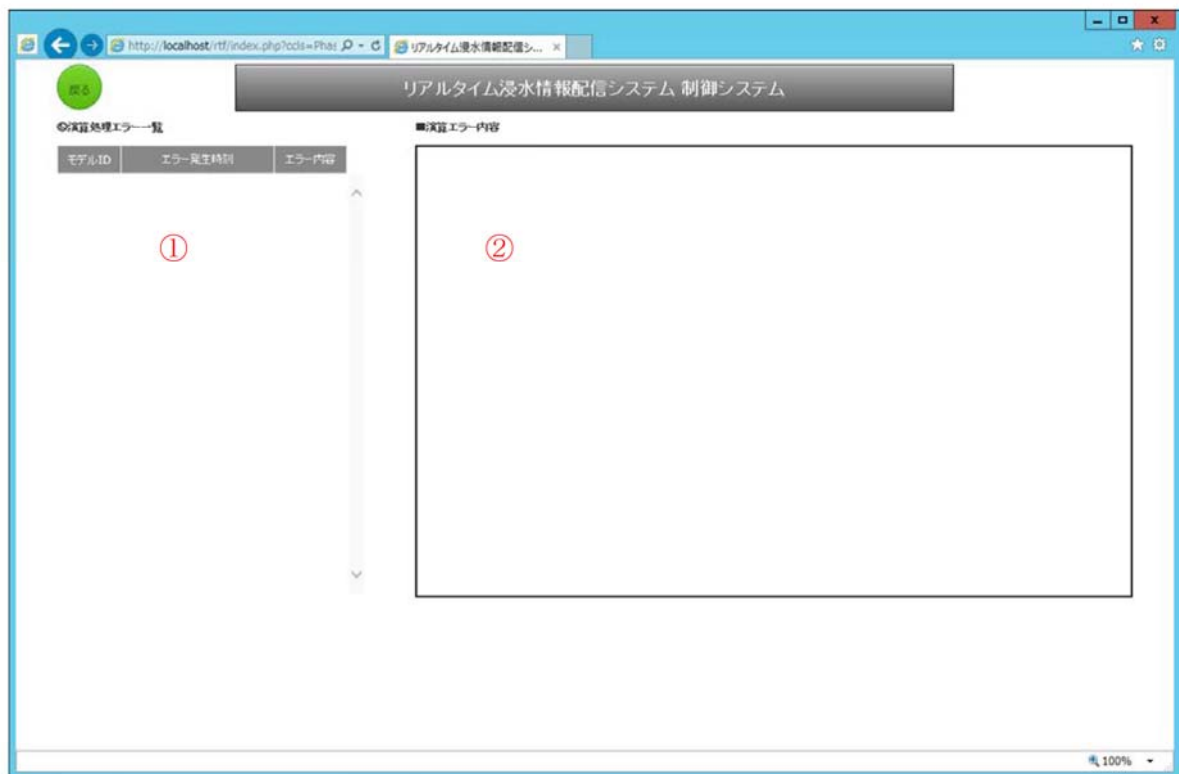


図 5-7 演算処理エラー一覧 画面

演算モデル（浸水予測システム）のエラー発生状況を表示

- ①エラー一覧は、モデル ID、エラー発生時刻、エラー内容を表示
- ②演算エラーの詳細を表示

5.8 配信用空間データベース格納処理

リアルタイム浸水情報配信システム 制御システム

配信用空間データベース格納処理

テレメータ雨量格納				テレメータ水位(河川)格納				メッシュ雨量(現況)格納			
格納時刻	格納完了時刻	処理時間	ステータス	基準時刻	格納完了時刻	処理時間	ステータス	基準時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:09	0秒	完了	2016-02-28 01:20	01:30:09	0秒	完了	2016-02-28 01:20	01:30:29	0秒	完了
2016-02-28 01:10	01:20:09	0秒	完了	2016-02-28 01:10	01:20:09	0秒	完了	2016-02-28 01:15	01:25:29	0秒	完了
2016-02-28 01:00	01:10:19	0秒	完了	2016-02-28 01:00	01:10:19	0秒	完了	2016-02-28 01:10	01:20:29	0秒	完了
2016-02-28 00:50	01:00:19	0秒	完了	2016-02-28 00:50	01:00:19	0秒	完了	2016-02-28 01:05	01:15:29	0秒	完了

メッシュ雨量(予測)格納				浸水深(現況)格納				浸水深(予測)格納			
基準時刻	格納完了時刻	処理時間	ステータス	基準時刻	格納完了時刻	処理時間	ステータス	基準時刻	格納完了時刻	処理時間	ステータス
2016-02-28 01:20	01:30:29	0秒	完了	2016-02-28 01:20		遅延		2016-02-28 01:20		遅延	
2016-02-28 01:15	01:25:29	0秒	完了	2016-02-28 01:10	01:24:42	0.6秒	完了	2016-02-28 01:10	01:24:43	0.6秒	完了
2016-02-28 01:10	01:20:29	0秒	完了	2016-02-28 01:00	01:14:32	0.6秒	完了	2016-02-28 01:00	01:14:33	0.6秒	完了
2016-02-28 01:05	01:15:29	0秒	完了	2016-02-28 00:50	01:04:27	0.6秒	完了	2016-02-28 00:50	01:04:28	0.6秒	完了

メッセージ

正常稼働中です。

■凡例

- 正常稼働中
- 処理遅延中
- エラー発生中

図 5-8 配信用空間データベース格納処理 画面

①テレメータ水位（河川）格納

テレメータ水位（河川）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②メッシュ雨量（現況）格納

メッシュ雨量（現況）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

③メッシュ雨量（予測）格納

メッシュ雨量（予測）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

④浸水深（現況）格納

浸水深（現況）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑤浸水深（予測）格納

浸水深（予測）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑥計算水位（現況）格納

計算水位（河川）（現況）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑦計算水位（予測）格納

計算水位（河川）（予測）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑧下水管内水位格納

下水管内水位格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑨下水道水位（現況）

下水道水位（現況）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑩下水道水位（予測）

下水道水位（予測）格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑪稼働状況表示

⑫稼働状況色別表示の凡例

5.9 タイル化処理

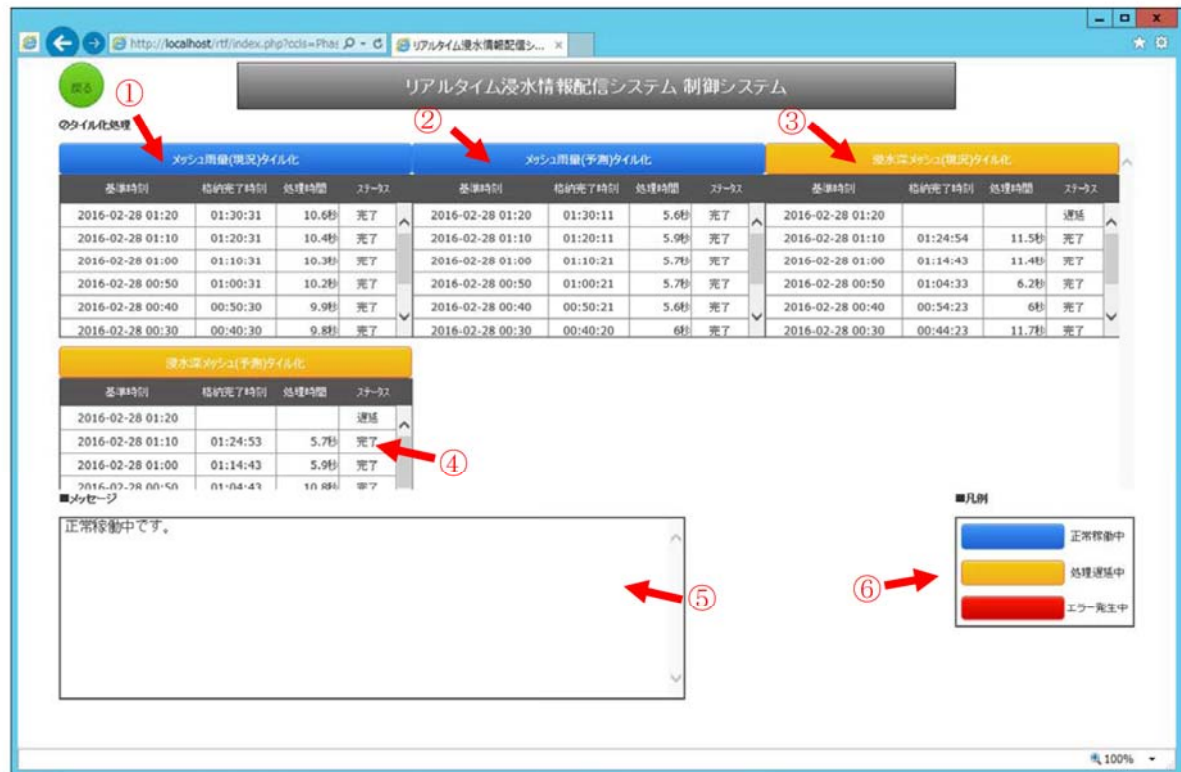


図 5-9 タイル化処理 画面

①メッシュ雨量（現況）タイル化

メッシュ雨量（現況）タイル化データ格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

②メッシュ雨量（予測）タイル化

メッシュ雨量（予測）タイル化データ格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

③浸水深メッシュ（現況）タイル化

浸水深メッシュ（現況）タイル化データ格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

④浸水深メッシュ（予測）タイル化

浸水深メッシュ（予測）タイル化データ格納の基準時刻、格納完了時刻、処理に要した時間、処理の状況（ステータス）をリアルタイムに表示

⑤稼働状況表示

⑥稼働状況色別表示の凡例

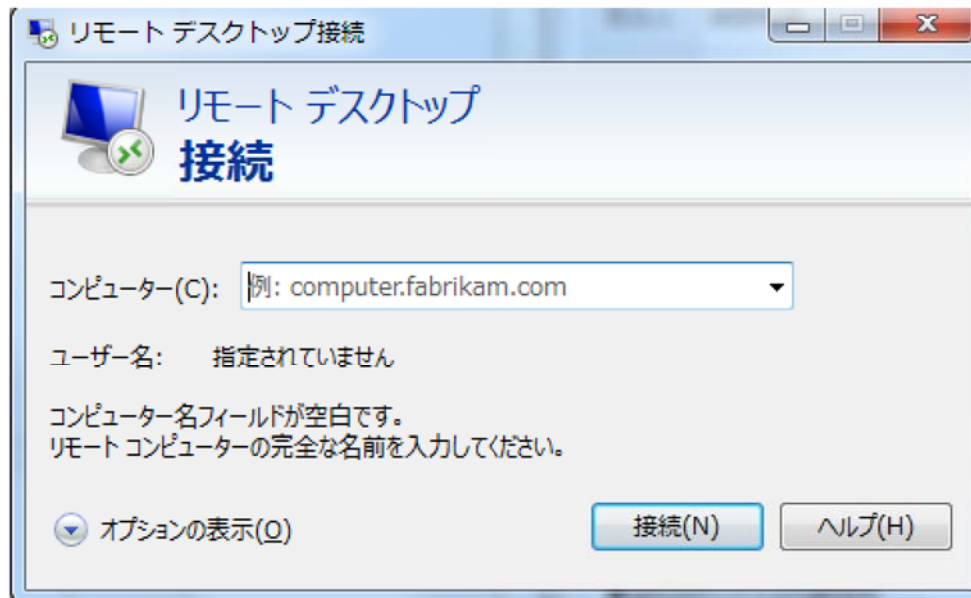
6. 浸水常襲箇所レイヤーの追加

①WEB サーバにリモートデスクトップ接続

VPN 接続を行った後、WEB サーバにリモートデスクトップ接続してください。

コンピュータ名 : XXXXXXXXXX

ユーザー名 : XXXXXXXXXX



②geojson ファイルの格納

作成したレイヤーを下記のディレクトリに格納してください。

/var/www/vhosts/shinsui/www/shinsui_archive/app/webroot/files/flood_jsk_area_json/

※レイヤーは geojson 形式で作成してください。

③設定ファイルの更新

「/var/www/vhosts/shinsui/www/shinsui_archive/app/webroot/js」ディレクトリに格納されている「index.js」を下記のように更新し、上書き保存してください。

1) 常襲エリアのレイヤー追加設定

「index.js」内を「常襲エリア」で文字列検索し下記の部分を追記します。

```

ファイル(F) 編集(E) 変換(C) 検索(S) ツール(T) 設定(O) ウィンドウ(W) ヘルプ(H)
0 1 2 3 4 5 6 7 8 9 10 11 12
527 cctv : new ol.source.Vector(),+
528 "hinanryo" : new ol.source.Vector(),+
529 "hinanbasyo" : new ol.source.Vector(),+
530 "suibo" : new ol.source.Vector(),+
531 "underpass" : new ol.source.Vector(),+
532 "chika" : new ol.source.Vector(),+
533 "kiken" : new ol.source.Vector(),+
534 "kanshi" : new ol.source.Vector(),+
535 "kanshi_over" : new ol.source.Vector(),+
536 "jiseki" : new ol.source.Vector(),+
537 "kuhochi" : new ol.source.Vector(),+
538 "ryuiki" : new ol.source.Vector(),+
539 "kadou" : new ol.source.Vector(),+
540 "rikukou" : new ol.source.Vector(),+
541 "flood_pnt" : new ol.source.Vector(),+
542 "f_area_jsk_01" : new ol.source.Vector(),+
543 "f_area_jsk_02" : new ol.source.Vector(),+
544 "f_area_jsk_03" : new ol.source.Vector(),+
545 "f_area_jsk_04" : new ol.source.Vector(),+
546 "f_area_jsk_05" : new ol.source.Vector(),+
547 "f_area_jsk_06" : new ol.source.Vector(),+
548 "f_area_jsk_07" : new ol.source.Vector(),+
549 "f_area_jsk_08" : new ol.source.Vector(),+
550 "f_area_jsk_09" : new ol.source.Vector(),+
551 "f_area_jsk_10" : new ol.source.Vector(),+
552 "f_area_jsk_11" : new ol.source.Vector(),+
553 "f_jyosyu_1" : new ol.source.Vector(),+ //常盤エリアのレイヤ生成+
554 };+
555 // add marker for each vectorLayer+
556 $.ajax({+
557   type: "POST",+
558   url: baseUrl+"map/ajaxUryo",+
559   data: {},+
560   dataType: 'json',+
561   success: function(data) {+
562     $.each(data, function(key, val) {+
563       var iconFeature = new ol.Feature({+
564         geometry: new ol.geos.Point(ol.proj.transform([parseFloat(val.lon), parseFloat(val.lat)], proj4326, proj3),+
565         name: val.name,+
566         //content: val.content,+
567       });+
568       vectorLayer["uryo_kansoku"].addFeature(iconFeature);+
569     });+
570   },+
571   error: function() {+
572     // 201807 DEL+
573     //alert('ajaxUryo');+
574   }+
575 });+
576 +
577 $.ajax({+
578   type: "POST",+
579   //url: baseUrl+"map/ajaxSui",+
580   url: baseUrl+"map/ajaxSuiModel",+
581   data: {'hyoji_jikan':hyoji_jikan,'hyoji_jikan_cur':hyoji_jikan_cur },+
582   dataType: 'json',+
583   success: function(data) {+
584     $.each(data, function(key, val) {+
585       var iconFeature = new ol.Feature({+

```

```

"f_area_jsk_11" : new ol.source.Vector(),+
"f_jyosyu_1" : new ol.source.Vector(),+ //常盤エリアのレイヤ生成+
};+
// add marker for each vectorLayer+
$.ajax({+
  type: "POST",+

```

データを追加する場合
「f_jyosyu_*」: new ol.source.Vector(),」を下の行に追加する。
*は連番となるようにしてください。

④レイヤーの色設定

「index.js」内を「常襲エリア」で文字列検索し下記の部分を追記します。

```

1262     text: new ol.style.Text({
1263     ^   ^   ^
1264     ^   ^   ^   font: '13px, Calibri, sans-serif',
1265     ^   ^   ^   textAlign: 'center',
1266     ^   ^   ^   textBaseline: 'top',
1267     ^   ^   ^   fill: new ol.style.Fill({
1268     ^   ^   ^   ^   color: 'fff'
1269     ^   ^   ^   ^   }),
1270     ^   ^   ^   stroke: new ol.style.Stroke({
1271     ^   ^   ^   ^   color: 'fff',
1272     ^   ^   ^   ^   width: 2
1273     ^   ^   ^   ^   }),
1274     ^   ^   ^   });
1275     ^   ^   ^   });
1276     ^   ^   ^   });
1277     ^   ^   ^   });
1278     ^   ^   ^   map.addLayer(vectorLayer[aryFloodAreaJskName[i]]);
1279     ^   ^   ^   });
1280
1281     //常襲エリアのレイヤ設定
1282     var flood_area_style = new ol.style.Style({
1283     ^   ^   ^   stroke: new ol.style.Stroke({
1284     ^   ^   ^   ^   color: 'AA0000',
1285     ^   ^   ^   ^   width: 1
1286     ^   ^   ^   ^   }),
1287     ^   ^   ^   fill: new ol.style.Fill({
1288     ^   ^   ^   ^   color: 'rgba(255, 0, 0, 0.3)'
1289     ^   ^   ^   ^   }),
1290     ^   ^   ^   });
1291
1292     //常襲エリアのGISデータ (geojson ファイルを列記する)
1293     var aryJyosyuAreaJskson = ["f_jyosyu.geojson"];
1294     var aryJyosyuAreaJskName = ["f_jyosyu_1"];
1295
1296     for(i=0; i<aryJyosyuAreaJskName.length; i++){
1297     ^   ^   ^   vectorLayer[aryJyosyuAreaJskName[i]] = new ol.layer.Vector({
1298     ^   ^   ^   ^   source: new ol.source.Vector({
1299     ^   ^   ^   ^   ^   url: baseUrl+"files/flood_jsk_area_json/"+aryJyosyuAreaJskJsn[i],
1300     ^   ^   ^   ^   ^   projection: "EPSG:4326",
1301     ^   ^   ^   ^   ^   format: new ol.format.GeoJSON()
1302     ^   ^   ^   ^   ^   });
1303     ^   ^   ^   ^   },
1304     ^   ^   ^   ^   name: aryJyosyuAreaJskName[i],
1305     ^   ^   ^   ^   style: flood_area_style
1306     ^   ^   ^   ^   });
1307     ^   ^   ^   map.addLayer(vectorLayer[aryJyosyuAreaJskName[i]]);
1308     ^   ^   ^   });
1309
1310
1311
1312
1313
1314     /**
1315     * Elements that make up the popup.
1316     */
1317     var container = document.getElementById('popup');
1318     //var content = document.getElementById('popup-content');
1319     //var closer = document.getElementById('popup-closer');
1320
1321     /**

```

```

//常襲エリアのレイヤ設定
var flood_area_style = new ol.style.Style({
  stroke: new ol.style.Stroke({
    color: 'AA0000',
    width: 1
  }),
  fill: new ol.style.Fill({
    color: 'rgba(255, 0, 0, 0.3)'
  })
});

```

→ 枠線の色を HTML カラーコードで指定します。

→ 枠線の太さを数値で指定します。

→ レイヤーの色を RGB で指定します。

→ レイヤーの透過度を数値で指定します (1.0=100%透過)。

⑤geojson ファイルの読み込み設定の更新

「index.js」内を「常襲エリア」で文字列検索し下記の部分を追記します。

```

1262     text: new ol.style.Text({
1263       text: text1,
1264       font: '13px,Calibri,sans-serif',
1265       textAlign: "center",
1266       textBaseline: "top",
1267       fill: new ol.style.Fill({
1268         color: '#f00'
1269       }),
1270       stroke: new ol.style.Stroke({
1271         color: '#fff',
1272         width: 2
1273       })
1274     });
1275   });
1276   });
1277   });
1278   map.addLayer(vectorLayer[aryFloodAreaJskName[i]]);
1279 }
1280
1281 //常襲エリアのレイヤ設定
1282 var flood_area_style = new ol.style.Style({
1283   stroke: new ol.style.Stroke({
1284     color: '#A00000',
1285     width: 1
1286   }),
1287   fill: new ol.style.Fill({
1288     color: 'rgba(255, 0, 0, 0.3)'
1289   })
1290 });
1291
1292 //常襲エリアのGISデータ (geojsonファイルを列記する)
1293 var aryJyosyuAreaJskJson = ["f_jyosyu.geojson"];
1294 var aryJyosyuAreaJskName = ["f_jyosyu_1"];
1295
1296 for(i=0;i<aryJyosyuAreaJskName.length;i++){
1297   vectorLayer[aryJyosyuAreaJskName[i]] = new ol.layer.Vector({
1298     source: new ol.source.Vector({
1299       url: "files/flood_jsk_area_json/"+aryJyosyuAreaJskJson[i],
1300       projection: "EPSG:4326",
1301       format: new ol.format.GeoJSON()
1302     }),
1303     name: aryJyosyuAreaJskName[i],
1304     style: flood_area_style
1305   });
1306   map.addLayer(vectorLayer[aryJyosyuAreaJskName[i]]);
1307 }
1308
1309
1310
1311
1312
1313
1314 /**
1315  * Elements that make up the popup.
1316  */
1317 var container = document.getElementById('popup');
1318 //var content = document.getElementById('popup-content');
1319 //var closer = document.getElementById('popup-closer');
1320
1321 /**

```

```

//常襲エリアのGISデータ (geojsonファイルを列記する)
var aryJyosyuAreaJskJson = ["追加する geosjon ファイルのファイル名"];
var aryJyosyuAreaJskName = ["手順③のレイヤー名"f_jyosyu_連番"];

```

1)2)の 2 行を下の行に追加します。

- 1)var aryJyosyuAreaJskJson = ["追加する geosjon ファイルのファイル名"];
- 2)var aryJyosyuAreaJskName = ["手順③のレイヤー名"f_jyosyu_連番"];

以上