

Magic uniPaaS V1Plus トラブルシューティング ツール



Enabling Business with Superior Technology

本書および添付サンプル(以下、本製品)の著作権は、マジックソフトウェアジャパン株式会社(MSJ)にあります。MSJ の書面による事前の許可なしでは、いかなる条件下でも、本製品のいかなる部分も、電子的、機械的、撮影、録音、その他のいかなる手段によっても、コピー、検索システムへの記憶、電送を行うことはできません。

本製品の内容につきましては、万全を期して作成していますが、万一誤りや不正確な記述があったとしても、MSE (Magic Software Enterprises Ltd.)および MSJ はいかなる責任、債務も負いません。本製品を使用した結果、または使用不可能な結果 生じた間接的、偶発的、副次的な損害(営利損失、業務中断、業務情報の損失などの損害も含む)に関し、事前に損害の可能性 が勧告されていた場合であっても、MSE および MSJ、その管理者、役員、従業員、代理人は、いかなる場合にも一切責任を負い ません。MSE および MSJ は、本製品の商業価値や特定の用途に対する適合性の保証を含め、明示的あるいは黙示的な保証は 一切していません。

本製品に記載の内容は、将来予告なしに変更することがあります。

サードパーティ各社商標の引用は、MSE および MSJ の製品に対する互換性に関しての情報提供のみを目的としてなされるものです。一般に、会社名、製品名は各社の商標または登録商標です。

本製品において、説明のためにサンプルとして引用されている会社名、製品名、住所、人物は、特に断り書きのないかぎり、すべて架空のものであり、実在のものについて言及するものではありません。

初版 2010年11月26日

第2版 2011年2月8日

マジックソフトウェア・ジャパン株式会社

目次

第1章 はじめに	. 5
第2章 uniPaaS 実行エンジンの出力ログ	. 6
2.1. アクティビティ モニタとは何ですか?	. 7
2.2. Studio でアクティビティ モニタ画面を開くには?	. 8
2.3. アクティビティモニタの「フィルタ」とは何ですか?	10
2.4. Studio でロギングダイアログを開くには?	11
2.5. ロギングダイアログでフィルタを設定するには?	12
2.6. アクティビティ モニタをファイル出力させるには?	13
2.7. ゲートウェイ ログとは?	15
2.8. ロギングダイアログでゲートウェイログを設定するには?	16
2.9. MAGIC.INI でゲートウェイログを設定するには?	18
2.10. ログ出力の例	19
2.11. 実行エンジンエラーログ	22
2.12. ログ出力のレベルとは?	23
2.13. ログの「同期」とは?	24
2.14. RIA クライアントログとは?	25
第3章 リクエスタログ	27
3.1. リクエスタログにはどんな種類がありますか?	28
3.2 リクエスタログ設定時の注意占は?	30
3.3 MRB 通信ログの目的と設定は?	31
3.4 MRB イベントログとは?	33
3.5 uniPaaSサーバログの目的と設定は?	34
3.6 インターネット リクエスタ ログの目的と設定は?	35
(1) 1) 1) 1) 1) 1) 1) 1) 1) 1) 1) 1) 1) 1	36
41 コマンドラインリクエスタの日的と利用方法は?	37
1 ?	12
9.2. ノローカビーク (mnb ビーク) C PD/C C C よ 9 が 1	40
5 9 ♀ ♀ ∠ ∠ ∠ ∠ ↑ ↑ ↓ L L L L L L L L L L L L L L L L L	+0 //7
5.1. ダンノノアイルを取得するメイミングと力広は · · · · · · · · · · · · · · · · · · ·	47 10
5.2. リーハモンユールのランフノアイル取得時の注意事項	40
J.J. 英市松丁时にダブフラアイルを1FMでとるには、 (Windows VD たただ Windows Server 2002 の提合)	10
(Williuows AF ゐよび Williuows Server 2003の場合)	49
5.4. 英吊於」時にダンフノアイルを作成させるには? (Vista Windows 7. Windows Server 2002の担合)	E 1
(Vista、Windows /、Windows Server 2008 の場合)	51
5.5. ハノクアッフ時にダンフファイルを作成させるには? (Windows VD t) htt Windows Convert 2002の担合)	E J
(Windows AP あよい Windows Server 2003の場合)	53
5.6. ハンクアップ時にダンフファイルを作成させるには? (Vista Windows 7. Windows Conver 2000 の根へ)	
(Vista、Windows /、Windows Server 2008 の場合)	55
5.7. ダスクマネーシャにフロセス ID を表示させるには?	5/
5.8. タンノを取るフロセスのフロセス IV を特定するには?	59
第 6 早 UNIPaaS テハッカ	bU
6.1. テパッカを開始する	61
	62
6.3. 行の無効化	66
6.4. フレイクボイントと行の無効化についての注意	68
6.5. 項目一覧	69
6.6. ウォッチリスト	71
6.7. コールスタック	73
6.8. 実行コンテキスト	75
第7章 リモートデバッガ	77
7.1. リモートデバッガのしくみはどうなっていますか?	78

7.2. リモートデバッグにはどんな準備が必要ですか?	79
7.3. リモートデバッグはどのように設定しますか?	80
7.4. アクティビティモニタを表示するためだけに使うには?	84
7.5. アクティビティモニタの内容を変更するには?	85
7.6. リモートデバッグのセキュリティは?	86
第8章 DBMS のユーティリティ	89
8.1. Pervasive PSQLのツール	90
8.2. MS SQL Serverのツール	99
第9章 uniPaaS製品以外のツール	. 102
9.1. Windows タスクマネージャ	. 103
9.2. リソースモニタ	. 107
9.3. システムイベントログ	. 110
9.4. その他のシステムモニタツール	. 111
9.5. パケットキャプチャプログラム	. 120
9.6. キーボードマクロ	. 125
9.7. その他	. 128
第10章 プログラミングの小技	. 130
10.1. Logging 関数	. 131
10.2. FlwMtr 関数	. 133
10.3. 項目の表示プログラム	. 134
10.4. コマンドラインリクエスタのログ読み込みプログラム	. 137
10.5. アクティビティモニタ出力の読み込みプログラム	. 139

第1章 はじめに

Magic uniPaaSのアプリケーションが思った通りに実行しない場合に、原因を特定するために各種のトラブルシューティング用のツールを活用すると、効率よく調査することができます。

本書では、トラブルシューティングに役立つツールおよびプログラムの小技について説明します。

本書の読者は、uniPaaSの基本についてすでに十分に理解していることを前提としています。また、Windows、 DBMS、ネットワーク等について基本的な技術知識があることを前提としています。

<u>第2章 uniPaaS 実行エンジンの出力ログ</u>

本章では、uniPaaS 実行エンジンが出力するログについて説明します。

種類	出力内容	設定	GUI に表示	ファイルに出力
アクティビティ モニタ	uniPaaS 実行エンジンの実行をトレース	MAGIC.INI (ロギング設定)	0	0
ゲートウェイ ログ	ゲートウェイ uniPaaS 実行エンジンが DBMS に発行 ログ する SQL 文などをトレース		0	0
実行エンジン エラーログ	実行エンジン uniPaaS 実行エンジンのエラーを記録 エラーログ			0
RIA クライアント RIA クライアントモジュールの実行状況 I ログ をトレース		MAGIC.INI (Studio)/ .publish.html		0
サーバ通信 uniPaaS サーバと、MRB/リクエスタと ログ の通信状況をトレース		MGREQ.INI		0
MRB 通信ログ MRB と他モジュールとの通信状況をトレース		MGRB.INI		0
MRB イベント MRB の主要イベントを記録 ログ		(自動)		0
インターネット リクエスタログ	インターネットリクエスタの TCP/IP レベ ルの通信状況を記録	MGREQ.INI		0

全体像を、RichClient Server システムについて図示したものが、下図です。ここには、uniPaaS 製品モジュール 以外のソフトウェアが出力するログも参考のために記載されています。これらのログについては第8章「DBMS のユーティリティ」および第9章「uniPaaS 製品以外のツール」で説明します。



2.1. アクティビティモニタとは何ですか?

アクティビティモニタは、uniPaaS 実行エンジンのデバッグ機能であり、Magic アプリケーションの実行の様子をトレースします。これには、Magic のコマンド(セレクト、コール、項目更新等々)の実行が1ステップごとにトレース されるので、Magic アプリケーションが意図したように動作しない場合に、プログラムの実行の流れをチェックす ることができます。

アクティビティモニタの出力は、次のいずれかの方法で記録されます。

方法	説明	参照
アクティビティ モニタ	Studio で開発時にデバッグしている場合にだ け利用可能。Studio の「アクティビティモニタ」 画面を開いて表示させます。	2.2 Studio でアクティビティ モニタ画面 を開くには?
ログファイルに出力	すべての uniPaaS 製品で利用可能。 アクティ ビティモニタの内容を指定したファイルに書き 出します。	2.6 アクティビティ モニタをファイル出力 させるには?
リモート デバッガ	リモート デバッガ上の「アクティビティ モニタ」 画面に表示します。 すべての uniPaaS 製品で 利用可能です。	第7章「リモートデバッガ」で説明します。



Studio では、モニタ画面、 ログファイルの両方に出力 できる。 Client および Enterprise Server/RichClient Server では、 ログファイルだけに出力できる。

2.2. Studio でアクティビティ モニタ画面を開くには?

開発の段階で、Studio 製品を使って単体テストを行っているときには、Studio の「アクティビティ モニタ」画面を 表示させることにより、プログラムの実行の流れを見ることができます。

プロジェクトを開いた状態で、メニュー
 「表示(V) → アクティビティモニタ(M)」を選択します。



 右図のような画面が表示されます。 初期状態では内容は空白です。 また、Studioの画面の状態によっては、Studioのウィ ンドウにドッキングした状態で表示されることもありま す。

89	アイビンアイセニュ	*		×
	▶ Ⅲ.	🧶 🕅 🛛	μ ¹ α	
	****	****	**** コンテキスト 7081624642560 *****************	^
14:	:13:00.193	-	->>開始 ロード バッチ タスク ー 'メインプログラム (TravelAgency)'	
14:	:13:00.193		終了 タスクのロード	
14:	:13:00.193		開始 奴功前	
14:	:13:00.193		フロー - ブロック I=If [実行しません]	
14:	:13:00.209		70~ - 7°D₀⁄ N=End	
14:	:13:00.209	- 1	終了 50.0前	
14:	:13:00.209		レコート"読込	
14:	:18:00.224	н — — — — — — — — — — — — — — — — — — —	>> 情報 >> IO file 'SUPPORT¥html_std.jpn' cannot be OPE	
14:	:13:00.255		DBテーブルのオープン - ' Locations (₩,₩)'	
14:	:13:00.255	-	DBテープ作のオープシー ' Airline_Locations (W,W)'	
14:	:13:00.255		>>開始 ロード オンライン タスク ー '位置' 修正 モード	
14:	:13:00.255		終了 奴りのロート	
14:	:13:00.255	-	レート、読込	
14:	:13:00.271	-	遅延トランザクション書込 4	
14:	:18:00.271		7四 項目 C=カカム '位置コード'(通常モード前方)	
14:	:13:02.271	-	- >> イベント処理中 [コンテキストがフォー加喪失(L)] コントロール名: [
14	10.00 400	-	N Ashikan methil Trippikoheriki makatul 🖉 🔹 🧯	×

アクティビティモニタにはツールバー(右図)があり、次ページの表のような機能を持っています。

これらの機能は、アクティビティモニタのコンテキストメ ニュー(右図)からも実行できます。

⊨ ∕∕	保存(S) クリア(<u>C</u>)
	検索(F) 再検索(N)
Ťan	開始/旅了ナエツ(<u>M</u>)
••••••••••••••••••••••••••••••••••••••	開始/於「フェッハM」 出力開始(0) 出力停止(U)

📙 🕨 💷 🖉 Ѩ 🧯 🛍

アイコン	コンテキストメニュー	機能
	保存(S)	ログの内容を、テキストファイルに書き出します。
•	出力開始(O)	ログ出力を開始します。
11	出力停止(U)	ログ出力を中断します。
	クリア(C)	ログ内容を消去します。
郤	検索(F)	ログの内容から文字列検索を行います。
Ţœ	再検索(N)	次の文字列検索を続けます。
100	開始/終了チェック(M)	対応する 開始 あるいは 終了 に移動します。



最後の開始/終了チェック(M)は、ロギング ダイアログ (2.5「ロギングダイアログでフィル タを設定するには?」参照)で、開始/終了メッ セージがYesになっている場合にだけ有効

です。

例えば、アクティビティモニタ上で、レコード後開始の行に カーソルがあった場合、開始/終了チェック(M) ボタンを押 すと、対応するレコード後終了の行にカーソルが移動し ます。

о п	ギング	X
設定	DBMS	
#	オフ。ション	
1	<u> </u>	Yes
2	1/*il	Yes
3	データビュー	Yes
4	再計算	Yes
5	処理フロー	Yes
6	ለኘንት	Yes
7	クライアントの動作	Yes
8	ゲートウェイ	Yes
9	トランサドクジョン キャッジュ	Yes
10	れ゛ックク゛ラウント゛ メッセーシ゛	Yes
11	開始/終了 メゎセージ	Yes 🔽
12	い。同期	Yes
13	外部ログファル名	mglog.log
		<u>ОК</u> + +>tell

2.3. アクティビティモニタの「フィルタ」とは何ですか?

アクティビティモニタの出力は、一番詳細に設定すると、各コマンドの実行、再計算などまで記録されるので、 簡単なプログラムの実行でも膨大な量になることがあります。あまり細かすぎると、重要な部分を見つけ出すた めに却って不便なことがあるので、必要な情報だけに取捨選択したいことがあります。このように出力情報を取 捨選択するために、ログメッセージの種類を指定することを「フィルタ」と呼びます。

アクティビティモニタの出力内容をフィルタするには、次の三つの方法があります。

方法	説明	参照
ロギングダイアログ	ロギングダイアログを開いて、フィルタ項目を 設定します。Studio でテスト実行する場合に だけ有効です。	2.4 Studio でロギングダイアログを開 くには?
MAGIC.INI	MAGIC.INI の [MAGIC_LOGGING] セクション でフィルタ項目をパラメータで指定します。す べての uniPaaS 製品で有効です。	2.6 アクティビティ モニタをファイル出 カさせるには?
Logging () 関数	uniPaaSプログラムの実行中に、フィルタする 内容を指定して Logging() 関数を実行します。 アプリケーション実行中に、プログラムで動的 にログレベルを変更することができます。	10.1「Logging 関数」で説明します。

2.4. Studio でロギングダイアログを開くには?

「ロギングダイアログ」は、Studio製品でだけ利用できます。ロギングダイアログを使うと、アクティビティログに 出力するメッセージの種別を指定することができます。デバッグの目的に応じて、出力の詳細さを変更すること ができます。

ロギング ダイアログ は、メニュー オプショ ン(O) から 設定(S) → ロギング(O) と選択 します。



ロギング ダイアログには、設定 と DBMS という二つのタブがあります。

- ・ 設定 タブを開くと、フローモニタに表示するトレースのレベルを設定できます。
- DBMS タブには、現在ゲートウェイがロードされている DBMS の一覧が表示され、それぞれにログレベルを設定できるようになっています。

ロギング		🥏 ロギング	
設定 BMS		設定 (DBMS)	
# オフ°ション	値	# DBMS ロウ*レ^*ル	
1 920	Yes 🗸 🗸	1 Btrieve MEZARC	
2 1/11	Yes	2 MicrosoftSQLServer N=なし	
3 データビュー	Yes	3 Memory N=なし	
4 再計算	Yes		
5 処理フロー	Yes		
6 7 ^{4°} 2h	Yes		
7 クライアントの動作	Yes		
8 ヴートウェイ	Yes		
9 トランサ [®] クジョン キャッシュ	Yes		
10 バックグラウンド メッセージ	Yes		
11 開始/終了 メーセージ	Yes		
12 口5 同期	Yes		
13 外部ログ ファイル名	mglog.log 💽		V
	OK ++)tl	ОК	++>\till



DBMS タブの設定は、ゲートウェイログを出力する場合に使います。ゲートウェイログについては、 2.7「ゲートウェイ ログとは?」で説明します。

2.5. ロギングダイアログでフィルタを設定するには?

アクティビティモニタのフィルタを設定するには、ロギ ングダイアログの「設定」タブで、アクティビティモニタ に出力する情報の種類を設定します。次のようなオ プションがあります。

ロギング ダイアログで設定した内容は、 MAGIC.INI の [MAGIC_LOGGING] セク ションに保存されます(次節参照)。このた め、いったん Studio を終了しても、次回 Studio を起 動したときには、前回の設定が記憶されています。

ę) []	シク	X
ſ	設定	BMS	
	#	オフ・ション	値
	1	920	Yes
	2	1/^*II	Yes
	3	データビュー	Yes
	4	再計算	Yes
	5	処理フロー	Yes
	6	ለጎንኮ	Yes
	7	クライアントの動作	Yes
	8	ゲートウェィ	Yes
	9	トランサドクジョン キャッシュ	Yes
	10	れ゛ックク゛ラウント゛ メッセーシ゛	Yes
	11	開始/終了 メールージ	Yes
	12	ゆ"同期	Yes
	13	外部ログファル名	mglog.log 😪
			OK ++)t

オプション	出力される内容
タスク	タスクの開始、あるいは終了時の、タスク名やタスクモード
レベル	タスク/レコード/コントロールの前処理/後処理などのハンドラが実行されたときの、レベル名やイベント名
データビュー	データ定義、範囲/位置付のような、データビューに関する情報
再計算	再計算処理時の変数名
処理フロー	処理コマンド名
イベント	トリガされたイベント名
クライアントの動作	リッチクライアントタスクにおける、クライアント側の動作のログ。 初期接続時にこのオプションが「No」に設定されている場合、あとから「Yes」に変 更してもクライアントにログが作成されません。 「Yes」に設定された場合、サーバとの更新回数が増えるので、パフォーマンスが 低下することがあります。
ゲートウェイ	ゲートウェイ ログの出力。ゲートウェイログを出力する場合、Yes にします。No にすると、「DBMS」タブでログレベルを設定してもアクティビティモニタには出力さ れません。
トランザクションキャッシュ	遅延トランザクション利用時の、トランザクションキャッシュへの操作。
バックグラウンドメッセージ	バックグラウンドメッセージがフィルタされるかどうかを指定します
開始/終了 メッセージ	ロギングモニタの処理の開始と終了の整合処理を有効にするかどうかを指定し ます
ログ同期	「外部ログファイル名」を指定することによりログをファイルに出力する際に、ログの同期を行うかどうかを指定します。ログの同期については、2.13「ログの「同期」とは?」を参照してください。 なお、「ロギングダイアログ」では、「フラッシュ」の指定はできません。
外部ログファイル名	ログ情報をファイルに書き込みたい場合、ファイル名を指定します。 フルパスを指定しない場合、アプリケーションの作業フォルダ上に作成されます。

2.6. アクティビティモニタをファイル出力させるには?

Studio 以外の製品では、アクティビティモニタ 画面が表示されないので、アクティビティモニタはファイル出力 だけができます。また、ロギング ダイアログが表示されないので、出力内容のフィルタの設定は MAGIC.INI の [MAGIC_LOGGING] セクションをテキストエディッタなどで直接編集して設定します。

アクティビティモニタをログファイルに出力するには、ログファイル名を MAGIC.INI の[MAGIC_LOGGING] で指定 します。同時に、ここにはフィルタの設定も行ないます。

以下の表は、フィルタのオプションと、MAGIC.INI で指定するパラメータ名、および各パラメータに指定可能な値を示したものです。

オプション	MAGIC.INI パラメータ名	設定可能な値
タスク	Task	Y/N
レベル	Levels	Y/N
データビュー	DataView	Y/N
再計算	Recompute	Y/N
処理フロー	Flow	Y/N
イベント	Events	Y/N
クライアントの動作	LogClient	Y/N
ゲートウェイ	Gateway	Y/N
トランザクションキャッシュ	TransCache	Y/N
バックグラウンドメッセージ	BackgroundMsg	Y/N
開始/終了 メッセージ	BeginEndMsg	Y/N
ログ同期	LogSynch	Y/N/F
外部ログファイル名	ExternalLogFileName	(ログファイル名)



- 外部ログファイル名(ExternalLogFileName)の指定は必須です。
- その他のパラメータは、指定されていなければデフォルト N とみなされます。
- ログ同期 (LogSynch) には、「Y」(同期あり)、「N」(同期なし)の他に、「F」(フラッシュ)の指定 をすることができます。(2.13「ログの「同期」とは?」参照)

設定例:次の設定例では、タスクレベルの出力のみを行い、ログ同期を行ないます。また、ログファイル名は mgmonitor.log であり、エンジンの作業ディレクトリ(アプリケーションの ECF ファイルの存在するディレクトリ) に作成されます。 [MAGIC_LOGGING] Task = Y Levels = N DataView = N Recompute = N Flow = N Events = N LogClient = N TransCache = N LogSynch = Y BeginEndMsg = N Gateway = N BackgroundMsg = N ExternalLogFileName = mgmonitor.log

2.7. ゲートウェイログとは?

Magic が DBMS にデータアクセスを行 うときには、その DBMS 用のゲート ウェイを通してアクセスします。従って、 Magic と DBMS との間のデータ・制御 のやりとりはすべて DBMS ゲートウェ イを通して行うことになります。

Magic 実行時に、予期しないレコード ロック、テーブルロック、ファイルマ ネージャの異常、パフォーマンスの問 題などが起こった場合には、Magic が DBMS をどのように利用しているのか、 あるいは、低レベルの DBMS とのイン ターフェースにおいて、どのようなデー



タやステータスコードが現れているのかなどを確認することが、問題解決につながることがあります。

Magic の DBMS ゲートウェイは、この目的のためのログ機能を備えています。

ゲートウェイログは、アクティビティモニタの機能の一環として実装されていて、アクティビティモニタの画面、あるいはログファイルに出力されます。

ゲートウェイログの設定は、アクティビティモニタの設定と同様に、次の三つの方法により行うことができます。

方法	説明	参照
ロギングダイアログ	ロギングダイアログを開いて、フィルタ項目を 設定します。Studio でテスト実行する場合に だけ有効です。	2.8 ロギングダイアログでゲートウェ イログを設定するには?
MAGIC.INI	MAGIC.INI の [MAGIC_DBMS] セクションでロ グレベルをパラメータで指定します。 すべての uniPaaS 製品で有効です。	2.9 MAGIC.INI でゲートウェイログを 設定するには?
Logging () 関数	uniPaaS プログラムの実行中に、フィルタする 内容を指定して Logging() 関数を実行します。 アプリケーション実行中に、プログラムで動的 にログレベルを変更することができます。	Logging 関数は、10.1「Logging 関 数」で説明します。

2.8. ロギングダイアログでゲートウェイログを設定するには?

Studio 製品上でテスト実行する場合には、ロギングダイアログを開いて、ゲートウェイログのログレベルを設定することができます。

 メニュー「オプション(0) → 設定(S) → ロ ギング(0)」を選択して、ロギングダイア ログを開きます。



 2.「設定」タブで、「ゲートウェイ」を Yes に 設定します。
 このオプションが No だと、ゲートウェイロ グは出力されません。

🥏 वर्म>ए	
設定 DBMS	
# オフ°ション	値
1 929	Yes
2 1/1	Yes
3 データビュー	Yes
4 再計算	Yes
5 処理フロー	Yes
6 /*')h	Yes
2 カライアントの通知作	Yes
8 5°~h9±1	Yes 🗸 🗸 🗸
<u>● トランサ[®]クジョン ちゃッジュ</u>	Yos
10 バックグラウンド メッセージ	Yes
11 開始/終了 メールージ	Yes
12 印"同期	Yes
13 外部ログ ファイル名	mglog.log
	OK ++)t

 「DBMS」タブで、ログをとりたいデータ ベースについて、「ログレベル」を設定し ます。
 右図では、MS SQL Server ゲートウェイ に対し、「S=サポート用」レベルのログを 出力することを指定しています。

🤣 ロギング	X
設定 DBMS	
# DBMS	בייע/יµ
1 Btrieve	N-tab
2 MicrosoftSQLServer	S=助称"~下用
3 Memory	NF 32 5 月 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1970年11 1070年11 1970 1970 1970 1970 1070 1070 1070 10
	OK (キャンセル

「ログレベル」としては、以下の指定ができます。「N=なし」以外では、C=ユーザ用、S=サポート用、D=開発者 用の順で出力量が多くなります。(2.12「ログ出力のレベルとは?」を参照)。

ログレベル	出力内容	出力量
N=なし	ログファイルが作成されません。	なし
D=開発者用	MSJの技術部門向けのログファイルが作成されます。	大
S=サポート用	アプリケーション開発者向けのログファイルが作成されます。	中
C=ユーザ用	SQLコマンドが出力するログのみ出力されます。	/]\

2.9. MAGIC.INI でゲートウェイログを設定するには?

Studio 以外の製品では、ロギングダイアログが使えないので、MAGIC.INIの指定により、ゲートウェイログの出力を指定することになります。

ゲートウェイログは、[MAGIC_DBMS] セクションで、DBMS の設定の一部(第5番目のパラメータ)として指定します。

例えば、MS SQL Server で「S=サポート用」レベルのゲートウェイ ログを出力させるためには、次のように指定します。

[MAGIC_DBMS]

MicrosoftSQLServer = 21, NotAllowNull, 10.3, MicrosoftSQL Parameters, **S**, + , NotLogSync, 0, 0, NotCheckExist,

ここで、「S」と指定されているものがゲートウェイログの指定で、「S=サポート用」レベルを意味しています。 「S」以外には、N(なし)、D(開発者用)、C(ユーザ用)が指定できます。

2.10. ログ出力の例

DBMS ゲートウェイは、DBMS ごとに内部処理が大きく異なりますので、ログも DBMS によりかなり異なるものとなりますが、特に、ISAM 系の DBMS ゲートウェイ (Pervasive, メモリなど)と、SQL 系の DBMS ゲートウェイ (Oracle、MS-SQL など)とは大きく異なります。

ここでは、詳しい解説はしませんが、以下に簡単なレコードフェッチのみを行った場合の例を挙げます。

2.10.1. ISAM 系 DBMS ゲートウェイの場合 (Pervasive)

Pervasive ゲートウェイのように、ISAM ファイルをアクセスするゲートウェイの場合には、DBMS が提供する API へのオペレーション、データ、戻り値などが記録されます。

赤色で示したところが、Pervasiveの提供する API (BTRV()と記されている)を呼び出している部分です。ファイルのオープン (OPEN)、STEP 命令や GET 命令などが発行されている様子がわかります。

```
17:49:03.546 - 3,91359 LogStart() : >>>> ***** Btrieve *****
17:49:03.546 - .91359 Version uniPaaS 1.8 SP1a, Log = mglog.log, Level = Support/QA, Sync = None, Date = 30/08/2010
17:49:03.546 - ,91359 LogStart() : <<<<<
17:49:10.000 - 0,97812 ==>_fil_open() open Counters (W,W,N) : normal (C:\Program Files\u00e4uniPaaS\u00e4Studio
V1Plus¥Projects¥TravelAgency¥Data¥)
17:49:10.000 - ).97812 ==> BTRV() #1 OPEN (0)
17:49:10.000
            .97812 >>
17:49:10.000 - ,97812 rec buf = 00
17:49:10.000 - .97812 <== OK (0)
17:49:10.000 - ,97812 ==> BTRV() #2 STAT (15)
17:49:10.015 – ,97828 <<
                          1. 2. 3. 4. 5. 6. 7. 8. 9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.
17:49:10.015
         17:49:10.015
         - ,97828 <== OK (0)
17:49:10.015 - ,97828 ==> BTRV() #3 STAT (15)
17:49:10.015 – ,97828 <<
                         1. 2. 3. 4. 5. 6. 7. 8. 9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.
17:49:10.015 – ,97828 ..... ( . . . .
                                17:49:10.015 - ,97828 <== OK (0)
17:49:10.015
         - ,97828 <== _fil_open() OK</p>
17:49:10.015
             ,97828 ==> _crsr_get() GET_GTEQ : crsr #0 (key #1, Counters)
17:49:10.015 -
             ,97828 key buf = 00 80
17:49:10.015 -
            .97828 ......
17:49:10.015 -
            ,97828 ==> BTRV() #4 GET_GE (9)
17:49:10.015 -
             .97828 <<
                           1. 2. 3. 4. 5. 6. 7. 8.
9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.36.37.38.39.40.
00 00 00 00 00 00 44 40
17:49:10.015 – ,97828 ...... . . Orders
                                                                          ....D@
17:49:10.015 - ,97828 <== OK (0)
17:49:10.015 - ,97828 ==> BTRV() #5 GET_POSITION (22)
17:49:10.015 - ,97828 <<
            ,97828 rec buf = 00 00 06 0a
17:49:10.015 -
17:49:10.015
             ,97828 <== OK (0)
17:49:10.015
              ,97828 <== _crsr_get() OK
17:49:10.015 - ,97828 ==> _crsr_set_curr() crsr #0 (key #1, Counters)
17:49:10.015 -
            ,97828 ==> BTRV() #6 GET_DIRECT/CHUNK (23)
17:49:10.015 -
             ,97828 >>
              ,97828 rec buf = 00 00 06 0a
17:49:10.015 -
         _
17:49:10.031
              .97843 <<
                           1. 2. 3. 4. 5. 6. 7. 8.
9.10.11.12.13.14.15.16.17.18.19.20.21.22.23.24.25.26.27.28.29.30.31.32.33.34.35.36.37.38.39.40.
00 00 00 00 00 00 44 40
17:49:10.031 – ,97843 ..... . . . Orders
                                                                          ....D@
17:49:10.031 -
             ,97843 <== OK (0)
```

17:49:10.031	-	,97843 <== _crsr_set_curr() OK	
17:49:10.031	-	,97843 ==>_crsr_get() GET_NEXT : crsr #0 (key #1, Counters)	
17:49:10.031	-	,97843 key buf = 01 00	
17:49:10.031	-	,97843	
17:49:10.031	-	,97843 ==> BTRV() #7 GET_NEXT_EXTENDED (36)	
17:49:10.031	-	,97843 >> 1. 2. 3. 4. 5. 6. 7. 8. 9.10.11.12.13.14.15.16.	
17:49:10.031	-	,97843 rec buf = 10 00 45 47 00 00 00 00 1e 00 01 00 28 00 00 00	
17:49:10.031	-	,97843	
17:49:10.031	-	,97843 <<	
17:49:10.031	-	,97843 rec buf = 00 00	
17:49:10.031	-	,97843 <== KNW_END_OF_FILE (9)	
17:49:10.031	-	,97843 <== _crsr_get() DB_ERR_NOREC	
17:49:11.500	-	1,99312 ==>_fil_close()	
17:49:11.500	-	,99312 ==> BTRV() #8 CLOSE (1)	
17:49:11.500	-	,99312 <== OK (0)	
17:49:11.500	-	,99312 <== _fil_close() OK	

2.10.2. SQL 系 DBMS ゲートウェイの場合 (MS-SQL)

SQL 系の DBMS ゲートウェイが出力するログでは、DBMS の提供する低レベルの API レベルでの操作よりも、 発行される SQL 文やその結果の方が重要です。

以下の例では、赤色で示した部分で、テーブルの存在確認、トランザクションの制御、レコード検索のための SELECT 文、データのフェッチなどが行われている様子が見て取れます。

17:54:11.093	-	1,98906	ms7_fil_open(): >>>>> ctxID = -1.000000, database = MAGIC, table = Counters, dbd->opened=0, share = W
17:54:11.093	-),98906 ı	ms7_connect_extra_session(): >>>>> database = MAGIC, pConnection->sess_extra = 0
17:54:11.093	-	,98906 ו	ms7_connect_extra_session(): >>>>> database = MAGIC, pConnection->sess_extra = 0
17:54:11.093	-	,98906 ו	ms7_esqlc_fil_exist() : db - MAGIC, owner - dbo, table - Counters
17:54:11.093	-	,98906 i	ms7_esqlc_fil_exist(): select id from MAGICsysobjects where name = 'Counters' AND uid = user_id('dbo') AND
(type = 'U' OR	type	e = 'V' OR	! type = 'S')
17:54:11.093	-	,98906 i	ms7_fil_open(): <<<<< ctxID = -1.000000, retcode = 0
17:54:11.093	-	,98906 ו	ms7_crsr_prepare(): >>>>> ctxID = -1.000000, database = MAGIC, key_idx = 0
17:54:11.093	-	,98906 ו	ms7_crsr_prepare(): table = Counters
17:54:11.093	-	,98906 ו	ms7_crsr_prepare(): number of blobs = 0
17:54:11.093	-	,98906 ו	ms7_connect_extra_session(): >>>>> database = MAGIC, pConnection->sess_extra = 0
17:54:11.093	-	,98906 i	ms7_crsr_prepare(): <<<<< ctxID = -1.000000, retcode = 0
17:54:11.093	-	,98906	ms7_crsr_begin(): >>>>> ctxID = -1.000000
17:54:11.093	-	,98906	ms7_crsr_begin(): <<<<< ctxID = -1.000000, retcode = RET_OK
17:54:11.093	-	,98906	ms7_trans(): >>>>> ctxID = -1.000000, transmode = 1, db = 20
17:54:11.093	-	,98906	Hold thru ms7_trans(): >>>>> ctxID = -1.000000, serverID = 3, dbconn_hdl = 1
17:54:11.093	-	,98906	Hold thru ms7_trans(): <<<<< ctxID = -1.000000, errcode = 0, RC = RC_OK
17:54:11.093	-	,98906	session – 0, in use – 1, write – 1, reusable – 1, results pending – 0,not only for cursors – 1
17:54:11.093	-	,98906	ms7_trans(): OPEN_READ
17:54:11.093	-	,98906	ms7_trans(): <<<<< ctxID = -1.000000, retcode = 0
17:54:11.093	-	,98906	ms7_crsr_open(): >>>>> ctxID = -1.000000, dbd_hdl = 0
17:54:11.093	-	,98906	ms7_crsr_open(): db = MAGIC, table = Counters, crsr_hdl = 0, dir = A, dir_rev = 0 , rngs = 0, Range:FALSE, Start
Pos:FALSE			
17:54:11.093	-	,98906	ms7_crsr_open(): checking if dummy fetch available
17:54:11.109	-	,98921	ms7_connect_extra_session(): >>>>> database = MAGIC, pConnection->sess_extra = 0
17:54:11.109	-	,98921	SET IMPLICIT_TRANSACTIONS ON
17:54:11.109	-	,98921	STMT: SELECT カウンターコート・カウンター説明,カウンター値 FROM MAGIC.dbo.Counters ORDER BY カウンターコート・ASC
17:54:11.109	-	,98921	STMT: SELECT カウンターコート・カウンター説明,カウンター値 FROM MAGIC.dbo.Counters ORDER BY カウンターコート・ASC
17:54:11.109	-	,98921	Using CURSOR
17:54:11.109	-	,98921	ms7_crsr_open(): <<<<< ctxID = -1.000000, retcode = 0
17:54:11.109	-	,98921	ms7_crsr_fetch(): >>>>> ctxID = -1.000000, dbd_hdl = 0, lock = FALSE
17:54:11.109	-	,98921	FETCH Read0
17:54:11.109	-	,98921	RESULT: 1
17:54:11.109	-	,98921	RESULT: Orders RESULT: 40.000000
17:54:11.109	-	,98921	ms7_crsr_fetch(): <<<<< ctxID = -1.000000, retcode = 0
17:54:11.109	-	,98921	ms7_crsr_fetch():
17:54:11.109	-	,98921	ms7_crsr_fetch(): <<<<< ctxID = -1.000000, retcode = 60
17:54:11.109	-	,98921	ms7_crsr_close(): >>>>> ctxID = -1.000000, crsr_hdl = 0

17:54:11.109	-	,98921 m	ns7_crsr_close(): <<<<< ctxID = -1.000000, retcode = RET_OK
17:54:11.109	-	,98921 m	ns7_trans(): >>>>> ctxID = -1.000000, transmode = 8, db = 20
17:54:11.109	-	,98921 se	ession – 0, in use – 1, write – 1, reusable – 1, results pending – 0,not only for cursors – 1
17:54:11.109	-	,98921 m	ns7_trans(): COMMIT
17:54:11.109	-	,98921 <mark>S</mark>	ET IMPLICIT_TRANSACTIONS OFF
17:54:11.109	-	,98921 m	ns7_trans(): <<<<< ctxID = -1.000000, retcode = 0
17:54:12.656	-	2,00468 m	ns7_crsr_end >>>>> ctxID = -1.000000
17:54:12.671	-	, 004 84 m	ns7_crsr_end <<<<< ctxID = -1.000000, retcode = RET_OK
17:54:12.671	-	, 004 84 m	ns7_crsr_release(): >>>>> ctxID = -1.000000
17:54:12.671	-	, 004 84 m	ns7_crsr_release(): database = MAGIC, table name = Counters, crsr_hdl = 0
17:54:12.671	-	, 004 84 m	ns7_crsr_release(): <<<< ctxID = -1.000000, retcode = 0
17:54:12.671	-	,00484 m	ns7_fil_close(): >>>>> ctxID = -1.000000, file : Counters, dbd->opened = 0
17:54:12.671	-	, 004 84 m	ns7_fil_close(): <<<< ctxID = -1.000000, retcode = 0

2.11. 実行エンジンエラーログ

2.11.1. 実行エンジンエラーログとは?

実行エンジンエラーログは、Magic アプリケーションサーバの実行中に重要な問題が起こった場合の、エラー情報を記録するものです。Magic プログラムがフォアグランドで動作している場合には、実行中にエラーが起こった場合に、ステータス行にもエラーメッセージが表示されます。

問題が起こった場合には、このログをチェックすることにより、問題についての情報を得ることができます。

具体的には、次のような情報が日時と共に出力されます。

- フォアグランドモードであればステータス行に表示されるようなエラー・ワーニングメッセージ。これには、 レコードやテーブルロック、DBMSのエラー、テンプレートファイルが開けない問題、UDF/UDPが見つ からない、その他、Magicの実行の継続ができなくなるようなエラーなどが含まれます。
- 「エラー」コマンドにより出力したメッセージ
- ライセンス関係のエラー。
- MRB やリクエスタとの通信中に起こった問題の記録。

実行エンジンエラーログは、プロジェクトを開く以前の問題(ライセンスの不正など)に対しては、uniPaaS製品を インストールしたディレクトリに出力されます。プロジェクトを開いて実行中のエラーに関しては、プロジェクトディ レクトリに出力されます。

2.11.2. 実行エンジンエラーログを出力させるには?

実行エンジンエラーログは、MAGIC.INIのパラメータ GeneralErrorLog で設定されます。デフォルトでは、 mgerror.log となっています。この設定を空白にすると、ログは出力されないようになります。

GeneralErrorLog = mgerror.log

2.11.3. ログファイル例

次に示すのは、エンジンエラーログの例です。各種のエラーが記録されているのがわかります。

24/06/2010 17:48:09.817	-	>> エラー >>>> 処理に失敗しました.他のユーザか処理にによってレコードが変更されています.データソース:
DUPREC_TBL2, program : de	eftrn	
24/06/2010 17:57:43.201	-	>> エラー >>>> オープンできません.データソース:DUPREC_TBL1, program : deftrn.create TBL1
24/06/2010 18:12:26.804	-	>> エラー >>>> インデックスが重複しています.データソース: DUPREC_TBL1, program : deftrn
24/06/2010 19:48:09.817	-	>> エラー >>>> インデックスが重複しています.テーブル: DENPYO
24/06/2010 20:57:43.201	-	>> エラー >>>> MRGSendResponse(): : "LOG_ON_1" ("KAISHA_SYS") (msgid 81)
24/06/2010 21:12:26.804	-	>> エラー >>>> ライセンス上の接続可能数を超えました.
24/06/2010 22:12:39.590	-	>> エラー >>>> インデックスが重複しています.テーブル: W_DENPYO
24/06/2010 23:12:41.529	-	>> エラー >>>> MRGSendResponse(): ERR-RUN-TIME-EXCEPTION(-139) : "MENU_H"
("KAISHA_SYS") (msgid 123	36)	
25/06/2010 01:48:09.124	-	>> エラー >>>> ユーザモジュールが見つかりません.
24/06/2010 02:57:43.567	-	>> エラー >>>> ユーザモジュールが見つかりません.
25/06/2010 08:12:26.436	-	>> エラー >>>> ユーザモジュールが見つかりません.
25/06/2010 12:12:39.742	-	>> エラー >>>> ユーザモジュールが見つかりません.
25/06/2010 16:12:41 589	_	>> エラー >>>> テンプレートファイルのオープン処理の問題です -

2.12. ログ出力のレベルとは?

ゲートウェイログ、あるいは後述のリクエスタログ(第3章「リクエスタログ」)では、ログの詳細さにより、 以下の3レベルがあります。(指定文字は、大文字でも小文字でも構いません)

指定文字	レベル	説明
С	簡易	ー番出力量が少ない設定です。オーバーヘッドも最小ですが、出力が少なすぎて必要な情報が記録されていないことがあります。
S	サポート	中間的な量のログが出力されます。ログの調査を行う場合には、最初はこのレベル が適当でしょう。
D	開発者	詳細なログが出力されます。通常はこのレベルのログは必要ありませんが、MSJで 詳細な調査を行う必要がある場合に、このレベルのログが必要になることがあります。

必要な情報の種類と、実行速度の許容範囲とから、適当なログレベルを選択してください。

一般的には、次のような選択になると思われます。

- 調査の初期段階で、「どのあたりに問題がありそうか?」について当たりをつける段階では、全体を見 渡しやすく、負荷も小さい C レベルを選択。
- 調査が進んでより深い情報が必要になってきたらSレベルを選択。
- MSJ のサポートセンターとのやりとりで必要になってきたら D レベルを選択。

2.13. ログの「同期」とは?

uniPaaSモジュールが出力するログで、「ログの同期」という設定がよく出てきます。これは、ログの各行をファイルに出力する際に、どのタイミングでディスクへの書き込みを行うか、を指定するもので、ここにはログの確実性(異常終了が起こった場合にログがどれだけ確実にファイルに書き込まれるか?)と、オーバーヘッドのトレードオフがあります。

2.13.1. ログの同期の種類

ログの同期には、以下の3種類があり、各ログの指定の箇所において、「設定文字」のような文字により指定されます。(設定文字は、大文字でも小文字でも構いません)

同期	設定文字
同期なし	N
同期あり	Y
フラッシュ	F

デフォルトの設定は「同期なし」であり、ログ出力は、uniPaaS モジュールのプロセスのメモリ内にバッファリング されます。これは C++言語のランタイムルーチンで自動的に行われている処理であり、ディスクへの書き込み のオーバーヘッドを最小にするために、一定量のデータが溜まるまでメモリ内に格納しておき、それを超えたら 初めてディスクへ書き込む、という処理を行ないます。

バッファリングを行うと、ログ出力に伴なうオーバーヘッド(処理速度の低下)を最小限に抑えることができますが、 問題点として、プロセスが異常終了した場合に、バッファリングされていてまだディスクに書き込まれていない データは、プロセスと一緒に失われてしまう、という点があります。異常終了の原因を追求するためにログの内 容を確認しようとしても、異常終了の直前の最も重要なタイミングでのデータが失われてしまって、原因追求が 思うようにできないことになります。従って、「同期あり」の設定は、異常終了しない場合の問題調査の場合に設 定するのが適当です。

「同期あり」にすると、バッファリングは行われず、1行出力するごとに、ファイルのオープン→書き込み→ファイ ルのクローズという処理が行われます。このような処理を行うことにより、ログの各行が確実にディスク上に書 き込まれるようになり、万一異常終了した場合でも、直前のログまでファイルに記録されるようになります。 一方、同期を行うことの問題点としては、1行ごとにファイルのオープン・クローズが行われるために、非常に オーバーヘッドが大きくなり、実行速度が遅くなることが挙げられます。実行速度が遅くなっても、確実にログを 記録しておいきたい、という場合には同期を行うように設定をしてください。

同期についての第三の選択肢として、「フラッシュ」という設定があります。これは、バッファリングは行わず、ロ グの行ごとにディスクにデータを書き込むようにするが、行ごとのファイルのオープン・クローズは行わない、と いうものです。この設定を行うと、異常終了時のログの確実性とオーバヘッドは、「同期あり」の場合と「同期な し」の場合の中間になります。

2.13.2. 同期の設定例

例として、アクティビティモニタのログ出力をMAGIC.INIで設定する場合(2.6「アクティビティモニタをファイル 出力させるには?」参照)には、次の行のようにします。これは、「同期あり」を指定した場合の例です。

LogSynch = Y

2.14. RIA クライアントログとは?

2.14.1. RIA クライアントログとは?

RIA クライアントログは、uniPaaS RichClient サーバシステムにおいて、クライアント側のモジュール uniRC.exe が出力する実行ログです。これには、クライアント側の通信内容、イベント処理、データやタスクの情報などの デバッグ情報が記録されます。

このログは、通常、uniPaaS アプリケーションの開発者が読んで解析する目的のものではありません。内部動作にかかわる記録がほとんどなので、MSJ のサポートセンターが問題の分析を行うためのものです。

2.14.2. RIA クライアントログの設定方法 (開発時)

開発時、Studioを使ってデバッグ実行する際には、RIA クライアントログの指定は、MAGIC.INI ファイルの [MAGIC_RIA] セクションで行ないます。

[MAGIC_RIA] ClientModulesPath=RIAModules¥uniRC_1_8_1_440¥ InternalLogLevel= InternalLogFile= InternalLogSync=Message DisplayStatisticInformation=N

このうち、以下のパラメータがログ設定に関連します。

パラメータ名	設定可能な値	説明
InternalLogLevel	SERVER	HTTPリクエストと応答のみ。
(ログレベル)	SUPPORT	SERVER 指定の内容に加え、サーバ間のデータの内容。
	GUI	クライアントによって記録された GUI メッセージの一部。
	DEV	クライアントによって記録されるすべてのメッセージ。
	(なし)	出力しない。
InternalLogFile (ログファイル名)	ファイル名	指定したファイルにログが出力されます。 ファイル名を指定しなければ、クライアントのデスクトップに uniRC_YYYY_MM_DD[.Process ID].log というファイル名で保存されます。
InternalLogSync	None	ログの各行について、ログファイルへの同期が行われます。
(ログの同期)	Session	ログファイルへの同期が行われません。このレベルは最も処理が早く なりますが、uniRC.exeの異常終了/ハングアップ時にログがすべて書 き込まれない可能性があります。
	Message	ログファイルは、各メッセージ毎にオープン/クローズされます。このレベ ルは、ログの保全性には優れていますが動作が遅くなります

2.14.3. RIA クライアントログの設定方法(実行時)

Studioを使わず、RichiClient Serverで RichClient プログラムを実行させる場合には、RIA クライアントログの設定は、起動用の.publish.html ファイル中で行ないます。

.publish.html ファイルというのは、uniPaaS Studio の「リッチクライアントインターフェースビルドダ」が作成する ファイルの一つで、デプロイメントマニフェストファイル .application ファイルと一緒に作成されます。作成場所は、 Studio をインストールしたディレクトリの下にある PublishedApplications¥(アプリケーション名)¥ サブディレクト リ中に作成されます。

ここで作成されるファイル名は、以下の二つがあります。

- (アプリケーション名).application … ClickOnce で利用するデプロイメントマニフェストファイル
- (アプリケーション名).publish.html … 起動用 HTML ページ

下図は、MyApp1という名前のアプリケーションについて作成されたファイルです。

😂 C:¥Program Files¥uniPaaS¥Stud	lio V1Plus¥PublishedAp	plications¥MyApp1	
ファイル(E) 編集(E) 表示(V) お気に入	り(A) ツール(エ) ヘルプ(日)		**
アドレス(D) 🛅 C:¥Program Files¥uniPaaS¥S	itudio V1 Plus¥PublishedApplio	cations¥MyApp1	🖌 🄁 移動
7 ส มชี ×	名前 🔺	サイズ 種類	更新日時
 ➡ Studio VI Plus ➡ Add_On ➡ Builders ➡ Cvs_Client ➡ Gateways ➡ license ➡ Projects ➡ Projects 	My App1 application My App1 publish html	6 KB Applicatio 10 KB HTML Doc 8 KB アイコン	2010/09/06 2010/09/06 2008/09/25
RIA Demo			

このうち、(アプリケーション名).publish.html ファイルをテキストエディタで開いてみると、次のような部分があります。このうち、key="InternalLogLevel"以下の3行で、ログについての指定を行ないます。

<xm l<="" th=""><th>id="rcExecProps"></th></xm>	id="rcExecProps">
	<properties></properties>
	<property key="protocol" val="http"></property>
	<property key="server" val="MyRCServer"></property>
	<property key="requester" val="/uni18Scripts/Mgrqispi018.dll"></property>
	<property key="appname" val="MyApp1"></property>
	<property key="prgname" val="MyRCProg1"></property>
	<property key="arguments" val=""></property>
	<property key="envvars" val=""></property>
	<property key="UseWindowsXPThemes" val="Y"></property>
	(途中省略)
	<property key="InternalLogLevel" val="server"></property>
	<property key="InternalLogFile" val="c:\tmp\rc.log"></property>
	<property key="InternalLogSync" val="Session"></property>
</td <td>xml></td>	xml>

この「val="…"」の部分に、それぞれ、ログレベル、ログファイル名、ログ同期の設定を指定します。指定する キーワードは、MAGIC.INIで設定する場合(2.14.2「RIA クライアントログの設定方法 (開発時)」)と同じです。

第3章 リクエスタログ

Magic で Web 対応アプリケーションや3階層アプリケーションを作成した場合には、クライアント側での Web ブラ ウザ、Magic クライアント、サーバ側での Web サーバ、Magic リクエストブローカ、Magic アプリケーションサーバ など、多くのソフトウェアコンポーネントが関連して実行しているので、問題の切り分けが複雑なります。根本的 な原因解明のためには、表面に現れる現象だけでなく、内部で起こっていることを正確に把握することが必要と なります。

Magic リクエスタログとは、Magic の Web アプリケーションや3階層アプリケーションに関係するモジュール間の 通信のトレースを保存する機能です。このログを見ることによって、Magic の Web アプリケーションにおいて障 害が起きた場合などに、原因の追求に役立ちます。ここでは、ログの種類と、その設定方法について説明しま す。

また、Magic のリクエスタログ機能の他に、Windows オペレーティングシステムやサードパーティのソフトウェア にも、プログラムの実行状況を監視するさまざまなユーティリティがあります。このような機能を併せて活用する ことにより、内部の動作を把握し、問題解決に至るまでの時間を大幅に短縮することができます。

3.1. リクエスタログにはどんな種類がありますか?

リクエスタログには、ログを出力するモジュールによって、数種類があります。最初に、リクエスタログを出力する uniPaaS モジュールについて説明します。

3.1.1. uniPaaS モジュールの種類

リクエスタログを出力する uniPaaS モジュールとしては、以下のようなものがあります。

モジュール名	ファイル名	ディレクトリ
uniPaaS サーバ実行エンジン	uniRTE.exe	uniPaaS ディレクトリ
リクエストブローカ (MRB)	uniRQBroker.exe	uniPaaS ディレクトリ
インターネットリクエスタ	MGrqispi018.dll MGrqcgi018.exe	Scripts ディレクトリ

ここで、インターネットリクエスタのファイル名にある「018」 は、Magic のバージョン番号で、バージョンが変わると番号 も変わります。

これらのモジュールは、右図のようにお互いに通信を行っており、それぞれの通信についてログが記録されることになります。



3.1.2. ログの種類

ログ出力が設定されていると、上記のそれぞれのモジュールが、他のモジュールとの通信を行うたびに、ログを 書き込みます。次の表は、ログの種類と、それを出力するモジュール名、およびログ出力を設定する INI ファイ ルとを示します。

ログファイル	出力するモジュール名	読み込む設定ファイル
MRB 通信ログ	リクエストブローカ (MRB)	uniPaaS ディレクトリの MGRB.INI
uniPaaS サーバログ	uniPaaS サーバ実行エンジン	uniPaaS ディレクトリの MGREQ.INI
インターネットリクエスタログ	インターネットリクエスタ	Scripts ディレクトリの MGREQ.INI

このうち、実際のデバッグで一番役に立つログは、MRB 通信ログです。MRB はすべてのリクエストの交通整理 を行い、また、uniPaaS サーバの実行状態を監視しているので、MRB のログを見ることにより、受け取ったリク エストの内容、リクエストを処理したエンジンの実行過程、エンジンの状態などを読み取ることができますので、 リクエストがどのように uniPaaS サーバで処理されたかを追跡するためには最適です。 uniPaaS サーバログおよびインターネットリクエスタログに記録されている内容は、TCP/IP に近い低レベルの 通信状況なので、ネットワークレベルでの接続の問題がある場合には原因究明のために使います。

ログをとるとそれなりのオーバーヘッドがありますので、目的に応じて選択してください。

3.2. リクエスタログ設定時の注意点は?

リクエスタログを出力するにあたっては、

- 出力ディレクトリに関する注意点
- パフォーマンスに関する注意点

があります。

3.2.1. ログ出力ディレクトリに関する注意点

ログの出力にあたっては、ログ出力先となるディレクトリを適当に決めます。ログ出力先のディレクトリについては、次の点に注意してください。

- 長い間ログ出力を行っていると、ログファイルの合計サイズが非常に大きくなるので、ログ出力先の ディレクトリは、十分な空き容量のあるドライブ上に指定する必要があります。ログのレベルやリクエス トの量にもよりますが、数時間で数百 MB くらいになることもあります。また、適時、不要なログは削除し てください。
- ディレクトリのセキュリティ権限の設定に気をつけてください。MRB や uniPaaS サーバをサービスとして 実行する場合には、実行ユーザ(デフォルトではローカルシステムアカウント)での作成・書き込み権 限が必要です。
- 同様に、インターネットリクエスタは、Web サーバと同じ Windows 資格情報で実行されますので、ログファイルを出力するディレクトリには、Web サーバを実行するユーザ での作成・書き込み権限が必要です。
- ログファイルは、決まった大きさ(数 MB くらい)になったら、自動的に分割され、日付と時刻などがファイル名に追加され、保存されます。(例えば、MRB.LOG が MRB_0210_1906.LOG などのようなファイル名に変更されます)。
- アクティブなログファイルにはなるべく触れないようにしてください。アクティブなファイルをテキストエディタで開いたり、別ディレクトリヘコピーなどをしたりすると、一時的ですが排他ロックがかかるので、それ以降のログ出力がされないようになってしまう場合があります。アクティブなログファイルとは、MRB.LOG などのように、「Log =」パラメータで指定されたファイル名のものです。一方、「MRB_0210_1906.LOG」などのファイル名を持つファイルは、サイズが大きくなったために分割・退避されたログファイルで、これはアクティブではありませんので、テキストエディタなどで開いても構いません。

3.2.2. パフォーマンスに関する注意点

リクエスタログを出力すると、パフォーマンスに影響が出ます。一般に、リクエストの量が多いシステムほど、また、パフォーマンスとのバランスについて、次の項目を実測しながら調節してください。

- ログ出力レベル: ログのレベルが高い(詳細なログ出力を行う)ほど、実行速度が遅くなります。2.12「ロ グ出力のレベルとは?」を参照して、必要最小限なログレベルを設定してください。
- ログ同期:異常終了はハングアップなど、実行モジュールの動作継続が不可能になってしまう場合のデバッグを目的とする場合には、同期=Yの設定が必要になりますが、オーバーヘッドが非常に大きいので、同期=フラッシュ(F)でもできないか、確認してください。実行モジュールの動作が継続する状況でのログ採取では、一般には同期=Nとするのが適当です。
- ログの種類: リクエスタログのうち、通常必要となるのは、「MRB 通信ログ」(3.3「MRB 通信ログの目的と 設定は?」)です。TCP/IP レベルでのエラーの確認が必要な場合にだけ、UniPaaS サーバログ(3.5 uniPaaS サーバログの目的と設定は?)やインターネットリクエスタログ(3.6 インターネット リクエスタロ グの目的と設定は?)をとってください。

3.3. MRB 通信ログの目的と設定は?

MRB 通信ログは、インターネットリクエスタからのリク エスト、uniPaaS サーバとの管理上の通信内容など、 MRB が行う通信内容を記録します。 MRB は uniPaaS サーバシステムの制御の要であるの で、MRB 通信ログを取得することにより、リクエストの 処理の流れを追っていくことができます。





MRB 通信ログを出力させるには?

MRB 通信ログを出力させるには、MRB モジュール (uniRQBroker.exe) が存在するディレクトリと同じディレクト リにある MGRB.INI ファイルで設定します。

- 1. Magic ディレクトリ中に、MGRB.INI というファイルがありますので、テキストエディタで開きます。
- 2. 「Log = ・・・」パラメータの行を探します。
- 3. デフォルトでは、行の最初にセミコロン「;」が入っていて、コメントアウトされていますので、セミコロンを はずします。
- 4. 「Log =」に続けて、ログファイル名、同期の有無、ログレベルを指定します。全体を括弧「(・・・)」でく くってください。
- 5. ファイルを保存します。
- 6. 設定を有効にするためには、MRBを再起動する必要があります。

例:

[MRB_E	NV]
Log =	(C:¥TMP¥LOG¥REQ.LOG Y S)

「Log =」に続くパラメータとしては、以下のものを空白文字で区切って指定します。

パラメータ順番	説明	例
1	ログファイル名 (混乱を避けるため、フルパスで指定す るのが良いでしょう)	C:¥TMP¥LOG¥REQ.LOG
2	ファイルの同期の有無を指定します。ログの同期につ いては、2.13 「ログの「同期」とは?」を参照してください。	Y(同期あり)
3	ログ出力のレベル: ログの詳細さを設定します。ログ出 カのレベルについては、2.12「ログ出力のレベルと は?」を参照してください。	S (サポート)

以下に、MRB 通信ログの例を示します。これは、MyApp1 という名前のアプリケーションを実行しているエン タープライズサーバで、MyProg1 という公開プログラム名のプログラムを実行させたときのログです。

, 2892,	2908	15:39:59,	, 28437	
, 2068,	2056	15:52:22,	, 71406	Version uniPaaS 1.8 SP1a, Log = C:\TMP\LOG\MRB.LOG, Level = Support/QA, Sync = Reopen,
Date =	= 01/09	/2010		
, 2068,	2056	,	, 71406	==> Version uniPaaS 1.8 SP1b Broker (build Mar 3 2010) : starting INITIALIZATION
, 2068,	2056	,	, 71406	BrokerPort = /5215
, 2068,	2056	,	71406	CommTimeout = 1000
, 2068,	2056	,	71406	ReLoad = TRUE
. 2068.	2056		71406	FloatingLicense = FALSE
. 2068.	2056		71421	Exe Entry : Online
. 2068.	2056		71421	command line : uniRTE.exe /DeploymentMode=R
. 2068.	2056	,	71421	start directory : C:¥Program Files¥uniPaaS¥Enterprise Server V1Plus
. 2068.	2056		71421	username. password :
. 2068.	2056		71421	on startup count : 0
. 2068.	2056		71421	on autoload count : 0
. 2068.	2056		71421	<== uniPaaS 1.8 SP1b Broker : INITIALIZED -OK (0)
, ,		, 		
2068	2056		71437	
2068.	2052	,	71437	
. 2068.	2228	15:52:27.	77093	==> mrb incoming msg()
. 2068.	2228	(5 sec).	77093	Message ENGINEOK (0)
. 2068.	2228		77093	From engine : MvServer/1501, 192,168.0.1 (pid 660)
. 2068.	2228		77093	INITIALIZATION,
. 2068.	2228		77093	Opened application : "MvApp1"
. 2068.	2228		77093	pool app insert ("MyApp1")
, 2068,	2228		77093	<== mrb_incoming_msg() INF-ACK_SENT (-44)
, 2068,	2228	,	, 77093	==> mrb_incoming_msg()
, 2068,	2176	15:52:37,	, 86796	==> mrb_incoming_msg()
, 2068,	2176	(10 sec),	, 86796	Message REQUEST SERVICE, -OK (0)
, 2068,	2176	,	, 86796	Regid : O
, 2068,	2176	,	, 86796	Program : MyProg1 ("MyApp1"), context -1, session 0
, 2068,	2176	,	, 86796	Priority : O
, 2068,	2176	,	, 86796	Client : 127.0.0.1 (pid 2232)
, 2068,	2176	,	, 86796	Filter :
, 2068,	2176	,	, 86796	==> Message ENGINE : MyServer/1501 (pid 660)
, 2068,	2176	,	, 86796	<== -0K (0)
, 2068,	2176	,	, 86796	$=> \log_update(1) IN_PROGRESS, 0, 0, 0$
, 2068,	2176	,	, 86796	<== log_update(1) 0 secs
, 2068,	2176	,	, 86796	<== mrb_incoming_msg() -OK (O)
, 2068,	2176	,	, 86796	==> mrb_incoming_msg()
, 2068,	2228	,	, 86828	Message ENGINE, -OK (0)
, 2068,	2228	,	, 86828	From engine : MyServer/1501, 192.168.0.1 (pid 660)
, 2068,	2228	,	, 86828	COMPLETED (Reqid 1) -OK (0)
, 2068,	2228	,	, 86828	$=> \log_u pdate(1) DONE, 0, 0, 0$
, 2068,	2228	,	86828	<== log_update(1) 0 secs
, 2068,	2228	,	, 86828	<== mrb_incoming_msg() -OK (O)
, 2068,	2228	,	86828	==> mrb_incoming_msg()
, 2068,	2176	15:52:41,	, 90796	Message END CONNECTION, INF-DONT-SEND-ACK (-21)
, 2068,	2176	(4 sec),	, 90796	<pre><== mrb_incoming_msg() INF-END-THREAD (-2)</pre>
, 2068,	2296	15:52:45,	, 95203	==> pool_delete() MyServer/1501
, 2068,	2296	(4 sec),	95203	$\leq = \text{pool}_{\text{delete}}() \forall \mathbf{W} (0)$
, 2068,	2228	,	, 95203	==> pool_delete() MyServer/1501
, 2068,	2228	,	95203	$\leq = pool_aelete()$ $\mathcal{I} \cdot \pi$
, 2068,	2228	,	95203	<pre><== mrd_incoming_msg() INF-ENU-IHKEAU (-2)</pre>
2060	2056	15.52.40	00220	
, 2008,	2000	10.02.49,	, ୭୬১८୪	

Г

3.4. MRB イベントログとは?

MRB イベントログというのは、MRB の活動の中で、特に重要なイベント(MRB の起動、終了、アプリケーション サーバの起動・登録・削除など)が記録されているログファイルで、MRB のモジュール(uniRQBroker.exe)が存 在するディレクトリに、mrb_event.log という名前で保存されます。

ここには、

- MRB の起動と終了
- [MRB_EXECUTABLES_LIST] に指定したプログラム自動起動
- Magic アプリケーションサーバの MRB への登録と登録削除(異常終了の場合も含む) などの情報が、日時、プロセス ID、その他のパラメータと共に記録されます。

Web 対応アプリケーションあるいはリモートコールを使うアプリケーションにおいて、Magic や MRB の動作に問題がある場合にここの記録を見ることにより、問題原因の手がかりを得ることができます。



MRB イベントログを出力させるための設定はありません。設定せずとも自動的に出力されます。逆にいうと、「出力させない」あるいは「ログレベルを変更する」ということはできません。

例:以下にmrb_event.logの例を示します。赤字で示したところは、Magicのアプリケーションサーバが異常終了したエラー記録の部分です。

1548 23:21:56,81718 28/08/2010 Startup (Version uniPaaS 1.8 SP1a, build Dec 2 2009)	
1548 ,81718 28/08/2010 BrokerPort = /5215	
1548 ,81718 28/08/2010 CommTimeout = 1000	
1548 , 81718 28/08/2010 ReLoad = TRUE	
1548 23:21:59, 84448 28/08/2010 Remote Spawn on "192.168.1.3/5225" ("MyApp1") : "OK" (0) (args: -	
[MAGIC_SERVERS]internal_broker=2, MyEntServer/5215,, [[[RemoteApp1]]]********, , 1 -MessagingServer=internal_broke	r
-ActivateRequestsServer=Y)	
1548 23:22:01,86975 28/08/2010 Remote Spawn on "192.168.1.3/5235" ("MyApp1") : "OK" (0) (args: -	
[MAGIC_SERVERS]internal_broker=2, MyEntServer/5215,, [[[RemoteApp3]]]********, , 1 -MessagingServer=internal_broke	r
-ActivateRequestsServer=Y)	
1548 23:22:03,89299 28/08/2010 Remote Spawn on "192.168.1.3/5245" ("MyApp1") : "OK" (0) (args: -	
[MAGIC_SERVERS]internal_broker=2, MyEntServer/5215,, [[[RemoteApp5]]]********, , , 1 -MessagingServer=internal_broke	r
-ActivateRequestsServer=Y)	
3544 ,89299 28/08/2010 Error: "TCP/IP: Connection reset" (-144) (/0)	
2880 23:22:04,89970 28/08/2010 Enterprise Server MyEntServer/1612 : Inserted (pid 2424 , UniCacklePlus ,	
license 0 threads 0 requests, "RemoteApp1")	
3424 23:22:06,91514 28/08/2010 Enterprise Server MyEntServer/1515 : Inserted (pid 4016 , UniCacklePlus ,	
license 0 threads 0 requests, "RemoteApp2")	
3244 23:22:07,93106 28/08/2010 Enterprise Server MyEntServer/1607 : Inserted (pid 3332 , UniCacklePlus ,	
license 0 threads 0 requests, "RemoteApp3")	
2380 23:32:41,26517 28/08/2010 Enterprise Servers shutdown:	
2380 ,26641 28/08/2010 Enterprise Server MyEntServer/1515 : Instructed to terminate : "OK" (0)	
2380 ,26641 28/08/2010 Enterprise Server MyEntServer/1515 : Removed : "OK" (0)	
3424 ,26719 28/08/2010 Enterprise Server MyEntServer/1515 : Notified termination	
2380 ,27016 28/08/2010 Enterprise Server MyEntServer/1612 : Instructed to terminate : "OK" (0)	
2380 ,27016 28/08/2010 Enterprise Server MyEntServer/1612 : Removed : "OK" (0)	
2880 ,27094 28/08/2010 Enterprise Server MyEntServer/1612 : Notified termination	
2380 ,27265 28/08/2010 Enterprise Server MyEntServer/1830 : Instructed to terminate : "OK" (0)	
2380 , 27265 28/08/2010 Enterprise Server MyEntServer/1830 : Removed : "OK" (0)	
2380 , 27265 28/08/2010 Broker shutdown:	
1548 23:32:48,33989 28/08/2010 Shutdown completed	

3.5. uniPaaS サーバログの目的と設定は?

uniPaaSサーバログは、uniPaaSサーバが出力するログファイルで、 インターネットリクエスタやMRBとの通信の状況を記録しています。 uniPaaSサーバ動作中に、接続が不安定になってエラーが発生す るなど、ネットワークレベルの問題が疑われる場合に収集して確認 します。



MRB 通信ログを出力させるには?

UniPaaS サーバログは、uniPaaS サーバモジュール uniRTE.exe が 存在するディレクトリ (uniPaaS のインストールディレクトリ) と同じ ディレクトリにある、MGREQ.INI ファイルで指定します。

1. uniPaaS ディレクトリ中に、MGreq.iniというファイルがありますので、テキストエディタで開きます。

WFF

ンターネット リクエスタ

uniPaaS

サーバログ

- 2. 「Log = ・・・」パラメータの行を探します。
- 3. デフォルトでは、行の最初にセミコロン「;」が入っていて、コメントアウトされていますので、セミコロンを はずします。
- 4. 「Log =」に続けて、ログファイル名、同期の有無、ログレベルを指定します。
- 5. ファイルを保存します。
- 6. 設定を有効にするためには、uniPaaSサーバを再起動する必要があります。

例:

[REQUESTER_ENV] Log = C:¥TMP¥LOG¥REQ.LOG Y S

「Log =」に続くパラメータとしては、以下のものを空白文字で区切って指定します。

パラメータ順番	説明	例
1	ログファイル名(混乱を避けるため、フルパスで指定す るのが良いでしょう)	C:¥TMP¥LOG¥REQ.LOG
2	ファイルの同期の有無を指定します。ログの同期につ いては、2.13 「ログの「同期」とは?」を参照してください。	Y(同期あり)
3	ログ出力のレベル: ログの詳細さを設定します。ログ出 カのレベルについては、2.12「ログ出力のレベルと は?」を参照してください。	s (サポート)

MRB

uniPaaS サーバ

3.6. インターネット リクエスタログの目的と設定は?

インターネット リクエスタ ログは、インターネット リクエスタが MRB や uniPaaS サーバにリクエスト処理を依頼するために行う通信内容 を記録するものです。

Web サーバ経由でクライアントから受け取ったリクエストを処理する際に、MRB や uniPaaS サーバとの接続が不安定になってエラーが発生するなど、ネットワークレベルの問題が疑われる場合に収集して確認します。



インターネット リクエスタ ログを出力さ せるには?



サーバ

- インターネットリクエスタのあるスクリプト ディレクトリの位置 を確認します。デフォルトでは、uniPaaS をインストールした ディレクトリの直下にある Scripts という名前のサブディレクトリです。
- 2. そのディレクトリにある MGREQ.INI ファイルを開きます。
- 3. 「Log = ・・・」パラメータの行を探します。
- 4. デフォルトでは、行の最初にセミコロン「; 」が入っていて、コメントアウトされていますので、セミコロンを はずします。
- 5. 「Log =」 に続けて、ログファイル名、同期の有無、ログレベルを指定します。 全体を括弧「(・・・)」でく くってください。
- 6. ファイルを保存します。
- 7. 設定を有効にするためには、IISを再起動する必要があります。

例:

[REQUESTER_ENV] Log = C:¥TMP¥LOG¥INETREQ.LOG Y S

「Log =」に続くパラメータは、uniPaaS サーバログの場合と同じです(3.5 「uniPaaS サーバログの目的と設定は?」参照)。



ここで指定するログファイル名は、Magic ディレクトリの MGreq.ini あるいは MGrb.ini に指定したの とは別のファイルを指定することをお勧めします。同じファイルを指定しても同期は取られますが、 多くのプログラムからのログが混在することになり、ログを読むのが難しくなってしまうからです。

<u>第4章 uniPaaS 製品添付のユーティリティ</u>

本章では、uniPaaS製品に添付されている各種ユーティリティの目的と使い方について説明します。

種類	出力内容	設定	GUI ツール	CMDツール
MRB/リクエストの状態	コマンドラインリクエスタ	(コマンドライン引数)		0
	MRB モニタ		0	
ライセンス利用状況 (FlexLM)	FlexLM ユーティリティ	Lmtools	0	
ライセンス利用状況 (uniPaaS)	ライセンス チェック ユー ティリティ	mgstations		0

このうち、FlexLMユーティリティについては、インストールガイドの「トラブルシューティング」に詳しい解説があるので、本書では省略します。

また、ライセンス チェック ユーティリティ (mgstations) については、リファレンスガイド 「ユーティリティ → MGSTATIONS」 に説明があるので、これも本書では省略します。
4.1. コマンドラインリクエスタの目的と利用方法は?

4.1.1. コマンドラインリクエスタとは?

コマンドラインリクエスタというのは、MRB に対してリクエストを発行するプログラムで、コマンドライン(DOS プロ ンプト)から実行するので、「コマンドライン」という名がついています。このプログラムは Magic をインストールし たディレクトリにインストールされ、名前は MGRQCMDL.EXE です。

コマンドラインリクエスタを使って、MRB モニタで表示されるのと同様な情報を表示させることができます。一般 には、MRB モニタは MRB に大きな負荷を掛ける傾向にあるのに対し、コマンドラインリクエスタは MRB にかけ る負荷が小さくすることができますので、MRB の状態を監視するには、MRB モニタを使うより、コマンドラインリ クエスタを推奨します。

コマンドラインリクエスタが持つ機能には多くのものがありますが、ここでは MRB の情報を取得するための基本的なコマンド使用法について説明します。詳しくは、Magic 添付のマニュアルを参照してください。

4.1.2. 現在実行中の uniPaaS サーバの一覧を見るには?

現在実行中(MRBの管理下にある)Magic アプリケーションサーバの一覧を見るには、「-query=rt」オプションを使います。

C∶¥Pr	rogram Files¥uniPaaS¥Enterprise S	Server V	1Plus> <mark>mgrqcmd</mark>	l -quer	y=rt		
	Enterprise Servers o	f (MyServ	ver/5215)				
#	EnterpriseServer	Pid	Status		Licen	ise	Application
1 2	MyServer/1501 192.168.0.1 MyServer/1797 192.168.0.1	3120 1648	Avail Idle Avail Idle	: 0 : 0	5 5	5 5	. 0 , 51 МуАрр1 , 0 , 49 МуАрр1

この例では、次のことがわかります。

- 二つのエンタープライズサーバが起動している。
- それぞれ MyApp1 というアプリケーションを実行している (Application 欄)

各行は、ひとつのエンタープライズサーバについての情報を表していて、それぞれ、次のことを意味しています。 (「値」は例の1行目)。

欄	値	意味
#	1	実行エンジンの番号
Enterprise Server	MyServer/1501 192.168.0.1	実行エンジンが実行されているホスト名と、エンジンの待ち 受けポート番号、IP アドレス。
Pid	3120	実行エンジンのプロセス ID。
Status	Avail Idle	実行エンジンの実行状態。
License	055,0,51	(後述)
Application	МуАрр1	実行エンジンが開いているアプリケーション名。

「License」欄の数字は、5つの数字(上記の例の1番目のサーバでは、「055,0,51」)からなっていますが、 それぞれ、次のことを意味しています。

欄	値	意味
1	0	現在実行中のスレッド数:実行中のスレッドはなし。
2	5	ピークスレッド数:最大5スレッドまで実行された。
3	5	ライセンスで割り当てられているスレッド数:5スレッドまで割り当てられている。
4	0	利用可スレッド数:0の場合には、割り当てられている最大スレッド数まで。
5	51	処理したリクエスト数:現在までに、51リクエストを処理した。

4.1.3. リクエストの統計情報を見るには?

リクエストによってマシンに掛かっている統計情報を見るには、「-query=load」オプションを利用します。

:¥Program Files¥uniPaaS¥Enterprise Server V1Plus> <mark>mgrqcmdl -query=load</mark>									
Statistics of (MyServer/5215, 10:51:24)									
Average Queue Time	Total	Pending	Executing	Completed	Failed				
0. 09	112	0	0	110	2				

この例では、次のことがわかります。

欄	値	説明
Queue Time	0.09	リクエストの平均待ち時間は 0.09 秒(すぐに処理されているということなので、 かなり余裕がある状態)
Total	112	今までに受け付けたリクエスト数
Pending	0	待ち行列に待たされているリクエスト数はなし。
Executing	0	現在実行中のリクエストはなし。
Completed	110	成功して完了したリクエスト数は 110。
Failed	2	何らかの理由で失敗したリクエスト数は 2。

4.1.4. 個々のリクエストについて表示させるには?

MRB内に記録されているリクエストを表示するには、「-query=log」オプションを利用します。ここに表示されるリクエストは、処理が完了(成功、失敗を問わず)したもの、処理中のもの、未処理のものすべてが含まれます。

ー般にはリクエスト数は何千何万という数になるので、出力するリクエスト数に制限をつけるのが一般的です。 ここでは、

- 最近の 20 個のリクエストだけを表示する
- リクエスト ID を特定して、一つのリクエストに関する情報だけを表示させる
- リクエスト ID の範囲を指定して、複数のリクエストに関する情報を表示させる

例1: 最近のリクエスト20 個の表示

最近の 20 個のリクエストを表示させるには、単に「-query=log」と指定します。リクエストの古い順に(リクエスト ID の昇順で)表示されます。

C:¥Program Files¥uniPa	aS¥Enterprise Ser	ver V1Plus	>mgrqcmdl -quer	y=log
Log of	requests (MySer	ver/5215,	16:38:46)	
# Request Status Id	Start Time	Elapsed (sec.)	Completion C (Mri, Runtime,	odes Dbms)
1 100 DONE Program : "e" ("MyApp Priority : 0	01/09 16:31:40 1") 3576) Enterpr	0	-OK (0) 0 0	
= = = = = = = = = = = = = = = = = = =	= = = = = = = = = = = = = = = = = = =	= = = = = = = =	-0K (0) 0 0	
Priority : 0 Client : MyServer (pid = = = = = = = = = = = ・・・ (以下省略)	3576), Enterpr = = = = = = = = = = =	iseServer = = = = =	: MyServer/1797 = = = = = = = = = =	

例2: リクエストID を指定して表示

調べたいリクエストのリクエスト ID がわかっている場合には、そのリクエスト ID を指定して表示させます。これには、「-query=log=(リクエスト ID)」というパラメータを使います。

;¥Program Files¥uniPaaS¥Enterprise Server V1Plus> <mark>mgrqcmd -query=log=132</mark>	
Log of requests (MyServer/5215, 16:52:40)	
Request Status Start Elapsed Completion Codes Id Time (sec.) (Mri, Runtime, Dbms)	
132 DONE 01/09 16:51:13 0 -OK (0) 0 0 Program : "e" ("MyApp1") Priority : 0 Glient : MyServer (pid 2420), EnterpriseServer : MyServer/1782	

例3: リクエストID の範囲を指定して表示

複数のリクエストを表示させるには、開始と終了のリクエスト ID をそれぞれ指定して、「-query=log=(開始リクエスト ID)-(終了リクエスト ID)」の形で指定します。

C:¥Program Files¥uniPaaS¥Enterprise Server V1Plus> mgrqcmdl -query=log=132-142								
_		Log o	f requests (rd200)3r2/5215,	16:54:44)			
#	Request Id	Status	Start Time	Elapsed (sec.)	Completion Codes (Mri, Runtime, Dbms)			
1 Pro Pri Cli	132 D ogram : " ority : O ent : rd2 = = = = =	00NE Ye″ (″MyAp 0 2003r2 (pi = = = = =	01/09 16:51:13 p1") d 2420), Enterpr = = = = = = = = =	3 0 iseServer	-OK (0) 0 0 : rd2003r2/1782			
2 133 DONE 01/09 16:51:13 0 -OK (0) 0 0 Program : "e" ("MyApp1") Priority : 0								
Cli = = •	ent : rd2 = = = = = • • (以	2003r2(pi ===== (下省略)	d 2420), Enterpr = = = = = = = = = =	iseServer = = = = = =	: rd2003r2/1907 = = = = = = = = = = = = = = = = = = =			



範囲を指定する場合、MRBに大きな負荷が掛からないように、必要最小限として、大きな範囲 を指定しないようにしてください。

4.1.5. 待ち行列中のリクエストを表示させるには?

MRB に記録されているリクエストのうち、まだ処理がされていないリクエストを表示させるには、-query=queue を指定します。

C:¥	Program Files¥u	ni PaaS¥Ent	erprise Server	V1Plus> <mark>mgrqc</mark>	mdl -query=que	eue	
	Ree	quests Que	ue of (192.168.	73. 136/5215,	16:59:42)		
#	Application	User	Prio Reqid	Client	(pid) S	Submit time	
1	МуАрр1			0 242	rd2003r2	3612	16:59:39
2	MyApp1			0 243	rd2003r2	3612	16:59:39
3	MyApp1			0 244	rd2003r2	3612	16:59:39
4	MyApp1			0 245	rd2003r2	3612	16:59:39

4.1.6. 実行可能なアプリケーションの一覧を表示させるには?

MRB の配下で実行している uniPaaS サーバが開いているアプリケーション名の一覧を表示させるには、- query=app を指定します。

4.1.7. 別マシンの MRB を指定するには?

今までの例では、同一サーバマシン上で実行している MRB に対してコマンドラインリクエスタを使っていました。 もし、別マシンに MRB がある場合、あるいは同一サーバ上でも別ポート番号で複数の MRB が動作している場 合には、MRB のホスト名・ポート番号を明示的に指定してやる必要があります。

デフォルトの設定 (つまり、MRB を明示的に指定しない場合)では、コマンドラインリクエスタは、カレントディレクトリ内にある MGREQ.INI に MessagingServer パラメータで指定されている MRB に対してリクエストを発行します。

例1:以下の MGREQ.INI の例は、MRB はホスト名 MyServer のサーバマシン上で、ブローカポート 5215 で実行している場合です。

[REQUESTER_ENV] Gateway = 1 MessagingServer = MyServer/5215

MRB が同一の PC 上で実行している場合には、ホスト名およびその直後のスラッシュ文字「/」は省略することができます。

例2:以下の例は、MRB が同一の PC 上で、ブローカポート番号 5215 で実行している場合の例です。

[REQUESTER_ENV] Gateway = 1 MessagingServer = 5215

もし、MGREQ.INI に指定してある MRB 以外の MRB を指定して、要求を発行するには、「-host=(ホスト名)」および「-port=(ブローカポート番号)」で指定します。

例 3: ホスト名 MyServer、ブローカポート番号 5215 で実行している MRB の配下で実行中のサーバエンジンの 一覧を表示させるには、次のようにします。

	Enterpri	se Servers of	(MySer	ver/5215)						
	EnterpriseSer	ver	Pid	Status		Licens	se			Application
 	MyServer/1501	 192. 168. 0. 1	3120	Avail Idle	: 0	0	 5	, 0, 0	===	====== MyApp1
2	MyServer/1797	192. 168. 0. 1	1648	Avail Idle	: 0	0	5	, 0 , 0		MyApp1

4.1.8. コマンドラインリクエスタのパラメーター覧を見るには?

コマンドラインリクエスタには、このほかにも多くのパラメータを指定できます。MGRQCMDL.EXE をパラメータなしで起動すると、下記のような簡単なヘルプが出てきます。

コマンドラインリクエスタの詳細な情報については、uniPaaS 製品のリファレンスマニュアルで「分散アプリケーション > アプリケーションパーティショニング > コマンドラインリクエスタ」を参照してください。

C:¥Program Files¥uniPaaS¥Enterprise Server V1Plus> mgrqcmdl Command Line Requester, Version uniPaaS 1.8 SP1a-0 12-345-6789
 -APPNAME -PRGNAME : Application and program names [-ARGUMENTS] : Program arguments, separated by commas [-VARIABLES] : Named variables, separated by commas [-PRIORITY] : Priority of execution (0 - 9) [-USERNAME] : Username required by the application [-PASSWORD] : Password required by the application or by the Broker [-FILENAME] : Name of a file to contain the request results [-NOWAIT] : Asynchronous request mode [-HOST, -PORT] : Broker's address
Data types : -A : alphanumeric -U : NULL -L : logical (True/False) -N : integer -F : float -D : double
Example : -APPNAME=Pet Shop Demo -PRGNAME=Orders List -PRIORITY=4 -USERNAME=supervisor -PASSWORD=mypass -FILENAME=MGRQCMDL.OUT -ARGUMENTS=string value, -N1000, -LTRUE, -U -VARIABLES=var 1=string value, var 2=-N1000, var 3=-LTRUE
-QUERY : Query requests - RT : [(appname)] Registered Enterprise Servers APP : [(host/port)] Applications supported by Enterprise Server(s) CTXS : (host/port) Contexts supported by Enterprise Server QUEUE : [(appname)] Requests in queue LOG : [(appname)][=reqid[-reqid]] Historic information LOAD : [(appname)] Statistics about application or the Broker PENDING : = <reqid> : Number of requests pending before the request -REQID : =<reqid> : Request manipulation (priority, removal) -CLEAR : Removing a request from the queue -EXE : =<exeentry>[/<args>] Activating an executable by the Broker -TERMINATE : Termination requests - ALL : all Enterprise Servers, including the Broker RTS : all Enterprise Servers, but not the Broker host/port : a specific Enterprise Server TIMEOUT : terminate the engine within this time period (seconds) Examples : -QUERY=RT -QUERY=RT (Pet Shop Demo) -QUERY=APP(my_server/1500)</args></exeentry></reqid></reqid>
Examples : -QUERY=RI -QUERY=RI (Pet Shop Demo) -QUERY=APP(my_server/1500) -QUERY=QUEUE -QUERY=LOG=100-90 -QUERY=LOAD (Pet Shop Demo) -EXE=Background//StartApplication=1 -TERMINATE=ALL -TERMINATE=my_server/1500 -CLEAR -REQID=1

4.2. ブローカモニタ (MRB モニタ)で何ができますか?

ブローカモニタ (MRB モニタ)は、ブローカの状態を監視するためのツールです。前述のコマンドラインリクエスタ が、コマンドラインからのツールであったのに対し、ブローカモニタは GUI を備えたツールとなっていて、一定間 隔で表示を自動的に更新します



ブローカモニタは便利なツールではありますが、定期的に多くの情報を取得するため、MRB にか かる負担は大きく、運用に支障を来たすことがあります。基本的に開発環境やテスト環境で利用し、 運用環境では常時使わないようにしてください。 ブローカモニタを常時利用している際に、サーバで不定期に問題が起こるようなことがあったら、ま

ずはブローカモニタの利用を停止して様子を見てください。



ブローカモニタをどうしても使いたい場合には、次の設定 によりMRBにかかる負担を軽減することができます。

- メニュー「オプション → 再表示設定」から、「再表示 間隔」を長くとる (例: 300 秒)。また、「表示リクエス ト数」を少なめにする (例: 30)
- 「コンテキスト」画面を閉じる。

再表示設定	×
再表示間隔と表示リクエスト数を設定して下さい。	
	_
再表示間隔(秒単位): 300	
表示リクエスト数: 30	
OKキャンセル	



ブローカモニタの使い方の詳細については、リファレンスマニュアル「分散アプリケーション → Broker モニタ」を参照してください。

4.2.1. 起動方法

ブローカモニタは、Windows の「スタート」メニューから「uniPaaS Enterprise Server V1Plus (インストールした uniPaaS 製品名) → ブローカとリクエスタ → Broker モニタ」を選んで起動します。

Administrator	🍓 Windows Update 😒 Windows カタログ	
🧊 サーバーの役割管理 🏠 エクスプローラ	m AMA Soft SnagIt 8	= • •
	📅 uniPaaS Enterprise Server V1 Plus 🛅 uniPaaS Studio V1 Plus	
	アクセサリ スタートアップ	
	管理ツール Immediator	▶ 愛 Magic uniPaaS VI Plus のヘルプ 🏾 🗽 MRBの起動 愛 Magic uniPaaS VI PlusのReadMe 🏾 🌆 MRBの停止
すべてのプログラム(<u>P</u>) ▸	 Ø Internet Explorer Outlook Express リエート アシッフタンフ 	MAGIO INI の編集 の uniPaaS VIPlus Broker INI の編集 の uniPaaS VIPlus Broker INI の編集
		оли муш
🎥 スタート 📔 🏉 🞯 🚱		

起動直後には、次のような画面が表示されます。

② Broker Monitor - モニタ中: Local Broker(localhost/5215) モニダ(M) 表示(V) 動作(A) オフジョン(Q) ウィントウ(W) ヘルフで出)		<u>_ 8 ×</u>
<u>بالمحمد المحمد المحم</u>	☞アフᡃᡃᢆᠨᢧ᠆ションサーバ	×
# 77° ½7 ½½ 送(信時初) 経過時間 ● 105 M M DONE 01/09 17:31:29 0 104 M FAILED 01/09 17:31:27 0 103 M FAILED 01/09 17:31:27 0 102 M FAILED 01/09 17:31:27 0 101 M M FAILED 01/09 17:31:8 0 100 M DONE 01/09 17:31:8 0 0 100 M M DONE 01/09 17:31:05 0 99 M M DONE 01/09 17:31:05 0 97 M M DONE 01/09 17:31:05 0 96 M M DONE 01/09 17:31:05 0 95 M M DONE 01/09 17:31:05 0		<u>現在</u> 0
<u>, ארי ארי ארי ארי ארי ארי ארי ארי ארי ארי</u>	 	×
	失敗したりガスト (3) 完了したりガスト (102) 処理中のりりガスト (0)	
	(0)	
	金リバスト の 50 100 150 200	250
	·	

4.2.2. アプリケーションサーバ

現在この MRB の配下で実行中のアプリケーションサーバ (Enterprise Server、あるいは Richclient Server)の 一覧を表示します。

コマンドラインリクエスタでの -query=rt に相当します。

ſ	7	ワリケーションサーバ											×
	#	:	PID	77%/5	状態	現在の	ヒペークス	最大スレット数	コンテキスト数	リクエスト数		その他	
		MyServer/1501	2344	My App 1	Avail Idle	0	5	5	0	51	0.002333		
	2	MyServer/1594	2412	Му Арр I	Avail Idle	U	5	5	U	БÜ	U JU4667		
Ш	1												

4.2.3. リクエスト

この MRB が受け付けたリクエストの一覧を表示します。

コマンドライアンリクエスタでの -query=log に相当します。

(🔵 ሃሳ፲ 7	۲ ト						×
IF	_			1			1	
Ш	#	アフツケーション	<u> ウライアント</u>	状態	送信時刻	経過時間	処理サーバ	▲
	106	MyApp1.e	MyServer	DONE	07/09 10:51	0	MyServer/1501	
Ш	105	MyApp1.e	MyServer	DONE	07/09 10:51	0	MyServer/1594	
Ш	104	MyApp1.e	MyServer	DONE	07/09 10:51	0	MyServer/1501	
Ш	103	MyApp1.e	MyServer	DONE	07/09 10:51	0	MyServer/1594	
Ш	102	MyApp11.e1	MyServer	FAILED	07/09 10:50	11		
Ш	101	MyApp1.e1	MyServer	FAILED	07/09 10:50	0	MyServer/1594	
Ш	100	MyApp1.e	MyServer	DONE	07/09 10:50	0	MyServer/1594	
Ш	99	MyApp1.e	MyServer	DONE	07/09 10:50	0	MyServer/1501	_ 1
	1.00	MUNITER -	Mi.C.	DONE	07/00 10.50	0	M.C. AEOA	

4.2.4. 統計

この MRB が受け付けたリクエストの状態についての統計グラフを表示します。 コマンドラインリクエスタでの -query=load に相当します。



4.2.5. 利用可能アプリケーション

この画面は、「アプリケーションサーバ」画面と連動しています。

「アプリケーション サーバ」画面に表示されている uniPaaS サーバのインスタンスをマウスクリックで選択します。 すると、そのアプリケーションサーバで実行されているアプリケーションが「利用可能アプリケーション」画面に 表示されます。

コマンドラインリクエスタでの -query=app に相当するものです。

Ӭ҄アフฃケーションサーバ							
#	77%ታ	│状態	現在の		最大スレット数	コンテキスト数	リクエスト数
1 MyServer/1501 2344 2 MyServer/1594 2412	My App1 My App1	Avail Idle Avail Idle	0 0	5 5	5 5	0 0	51 60
	()利用可能	アフリケーション	,				×
	1 My	App1	MyServer/	/1501			

4.2.6. コンテキスト

「コンテキスト」画面も「アプリケーションサーバ」画面と連動しています。

「アプリケーション サーバ」画面に表示されている uniPaaS サーバのインスタンスをマウスクリックで選択します。 すると、そのアプリケーションサーバで保持されているコンテキストの一覧が表示されます。

🕗 アフ判ケーションサーバ									
# #74/#-	PID	アフジケーション	状態		現在(のス 【ピークス	い… 最大…	コンテキスト数	95
1 MyServer/1501 2 MyServer/1502	3840 3928	My App1 My App1	Busy R Busy R	equest equest	0	5	5	3 0	54 50
	コンテキスト								×
	עב 🛊	F#ZNID	 リクエス	ንግን	ב- <u></u>	状態	未使用時間		- 6
	1 456 2 465	574051614720 554321838080	103 104	ex ex		Pending Pending	00:01:22 00:01:20	_	
	3 4/4	134392061440	105	ex		rending	00:01:19		.1



ブローカモニタが MRB に大きな負荷を与えて動作を不安定にすることがあるので、「コンテキスト」 画面はできるだけ閉じておくことを推奨します。

第5章 ダンプファイル

ダンプファイルというのは、Windows上で実行しているプログラムについて、ある特定の時点でのプロセスのメ インメモリの状態と、プロセスに関連する情報をハードディスクに保存したものです。ダンプファイルの種類にも よりますが、プロセスの仮想空間のデータを保存するので、最大数百 MB にもなることがあります。

ダンプファイルは、プログラムの実行に異常(異常終了、ハングアップ、不明なエラーなど)があった場合に、原因を解析するのに非常に有効ですので、uniPaaSシステム実行中に問題が起きた場合に、MSJのサポートセンターからダンプファイルの取得を依頼することがあります。

本章では、ダンプファイルの取得の方法について説明します。



ダンプファイルの解析には、プログラムのソースコードをはじめとして、さまざまな内部情報が必要になるので、ダンプファイルの解析は MSJ において行ないます。uniPaaS アプリケーションの開発者が解析を行うことはできません。



ダンプファイルの作成は、Windowsの機能ですので、技術的詳細については、Microsoft 社の Web サイト、あるいはインターネットの検索で「Windows ダンプファイル」などをキーワードとして検索す れば多数見つけることができます。

5.1. ダンプファイルを取得するタイミングと方法は?

ダンプファイルは、プログラムの実行中にある一時点におけるプロセスの状態をファイルに保存するものですが、 効果的に解析するには、取得するタイミングが重要です。 一般には、問題が起こった時点でのプロセスの状態をダンプファイルにとります。例えば、

プログラムが異常終了する。(クラッシュ)

- プログラムが反応しなくなる。(ハングアップ)
- エラーダイアログが表示される。

などです。

ダンプファイルを取得するための方法は、Windowsのバージョンにより異なります。

- Windows XP および Windows Server 2003 では、ワトソン博士 を使います。
- Windows Vista、Windows Server 2008、Windows 7 では、タスクマネージャを使います。

以下、(1) 異常終了時にダンプファイルを作成させる方法、(2) ハングアップ時にダンプファイルを作成させる方法、のそれぞれについて、各 Windows バージョンに分けて説明します。

エラーダイアログが表示される場合には、ハングアップの場合と同じ方法により、ダンプファイルを作成することができます。

5.2. サーバモジュールのダンプファイル取得時の注意事項

uniPaaSのサーバ製品 (Enterprise Server、RichClient Server)をマルチスレッドのバックグラウンドモードで実行している場合には、以下のパラメータを MAGIC.INI に設定してください。

[MAGIC_SPECIALS] ExceptionMessageBoxDisplay = Y

このパラメータは、uniPaaS 実行エンジンのエラー処理の方法を制御します。

- Nの場合(デフォルト)では、uniPaaS実行エンジンがマルチスレッドでアプリケーションを実行中に、ある スレッドで実行を継続できなくなるような例外状態(メモリアクセス違反など)が発生した場合、そのスレッ ドは終了させるけれども、エンジン自体はリカバリー処理を行って、継続して他のスレッドの処理の実行を 続けます。これは、通常の運用時に適当な設定であり、一つのスレッドの異常が他のスレッドの実行に影 響を与えないようになり、可用性が向上します。
- Yの場合には、例外状態が発生した場合には、その後の処理はWindowsのデフォルトのエラー処理に 任せます。メモリアクセス違反などのOSレベルでの例外が起こった場合には、通常、Windowsはダンプ ファイルを作成してそのプロセスを停止します。このとき、同時に実行していた他のスレッドや、そのプロ セス(インスタンス)に格納されていたコンテキストなどは、プロセスの終了と同時に中断・破棄されます。こ のため、運用上の可用性は低くなります。

通常の運用時には N に設定しておくのが適当ですが、この設定では異常事態が発生した場合にリカバリ処理 が働いてダンプファイルが作成されず、「ダンプファイルを調査して原因を追求する」という方法が使えなくなっ てしまいます。このため、デバッグ時に限っては、クラッシュダンプを作成するために、Y に設定して、あえてプ ロセスを強制終了させるようにする必要があります。

ただし、この設定では、上記のように運用上の可用性が低くなるため、デバッグ時に限り設定するようにして、 調査が終わったら、元通り N にする (あるいはパラメータをコメントアウトする)ようにしてください。

5.3. 異常終了時にダンプファイルを作成させるには? (Windows XP および Windows Server 2003 の場合)

Windowx XP および Windows Server 2003 では、 ワトソン博士 drwtsn32.exe をデフォルトのデバッガとして Windows に登録しておくことにより、 プログラムが異常終了した際に自動的にダンプファイルを作成するように 設定しておくことができます。



- Administrator 権限のあるユーザとしてログインします。
- 5. コマンドプロンプト(DOS 窓)で、「drwtsn32 -i 」とタイ プします。右図のようなダイアログが表示されます。
- コマンドプロンプトから、今度は引数なしで 「drwtsn32」を実行すると、右図のような設定ダイア ログが表示されます。

次のような設定を行ってください

「ログファイルパス」のファイル名、および「クラッシュ ダンプ」欄のパス名は、十分な空き容量のあるディス ク上にあることを確認してください。

クラッシュダンプの種類: 完全 メッセージボックスによる通知: チェックする クラッシュダンプファイルの作成: チェックする

1		
	Dr. Watson	×
0	・リン博士が既定のアプリケーション デバッガとしてインストールされま	した。
	Windows ワトソン博士 ?×	
ア ユ (ス	D ^グ ファイル パス(L): ition Data¥Microsoft¥Dr Watson クラッシュ ダンブ(P): C×Documents and Settings¥Adı WAVE ファイル(F): 参照(W) インストラクションの数(D): 10 パインストラクションの数(D): 10 クラッシュ ダンブの種類(Y): 完全 最小 ○ 完全 (NT4 互換) オプション ダンプ シンボル テーブル(D) マ すべてのスレッド コンテキストをダンブ(Δ) マ 野なのロパ ファイル(ご良知(E))	
	▼ メッセージ ボックスによる通知(U) 音による通知(S)	
	マラッシュダンプファイルの作取(D)	
	アプリケーション エラー(<u>R</u>) <u> あ示(い)</u> <u> りリア(O)</u>	
	OK キャンセル ヘルプ(H)	

これで設定は終了です。この設定は Windows のレジストリに登録されますので、一度だけ行えば、その後ずっと有効です。

この設定を行って、実際にプログラムのテスト・運用を行ないます。実行中に、アクセス違反、ハンドルされない 例外などが発生してプロセスが異常終了したら、次の手順でダンプファイルを採取してください。



プログラムが異常終了した場合には・・・

 プログラムが異常終了した場合には、Windows は自動的にワトソン博士を起動し、右図のような ダイアログが表示されます。 ここのボタンが「キャンセル」になっている間は、 ログファイルを作成中ですので、ボタンを押さない でください。

 しばらくすると(最大1分程度)、ボタンが「キャン セル」から「OK」に変わります。
 ボタンが「キャンセル」から「OK」に変わったら、ロ グファイルおよびクラッシュダンプファイルの作成 は完了ですので、OKを押してワトソン博士を終 了させてください。



作成されたダンプファイルは、ワトソン博士の「クラッシュダンプ」欄に指定されたディレクトリに保存されています。デフォルトでは、Windows XP では

C:\Documents and Settings\All Users\Application Data\Microsoft\Dr Watson\user.dmp

あるいは、Windows Server 2003 では

C:\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Dr Watson\user.dmp

です。

5.4. 異常終了時にダンプファイルを作成させるには? (Vista、Windows 7、Windows Server 2008 の場合)

Windows Vista、Windows 7、Windows Server 2008 の場合には、ワトソン博士はなく、レジストリにデフォルトの デバッガを設定します。



ダンプファイル作成の設定を登録するには

1. レジストリエディッタを開いて、次のようにキーを設定します。

キー:HKEY_LOCAL_MACHINE¥Software¥Microsoft¥Windows¥Windows Error Reporting¥LocalDumps

ダンプファイルを格納する場所	値の名前:	DumpFolder
	種類:	REG_EXPAND_SZ
	値のデータ:	c∶¥CrashDumps
上記場所に格納できるダンプファイ	値の名前:	DumpCount
ルの最大数。	種類:	DumpCount REG_DWORD Oxa
	値のデータ:	Оха
ダンプのレベル	値の名前:	DumpType
	種類:	REG_DWORD
	値のデータ:	0x2

ここで、「ダンプファイルを格納する場所」が c:¥CrashDumps になっていますが、任意の場所で構いません。空き容量が十分にあるハードディスク上のディレクトリを選択してください。



レジストリエディタを使わず、.reg ファイルから登録することもできます。 上の設定のままならば、次のような内容を tmp.reg など(名前は任意で構いませんが、拡張子は .reg とします)に保存してから、ダブルクリックしてレジストリに登録してください。

Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Windows Error Reporting\LocalDumps]
"DumpFolder"=hex(2):63,00,3a,00,5c,00,43,00,72,00,61,00,73,00,68,00,44,00,75,\
00,6d,00,70,00,73,00,000
"DumpCount"=dword:0000000a
"DumpType"=dword:00000002

設定はこれだけで、すぐに有効になります。

この設定を行ってから、uniPaaSアプリケーションをテスト実行・運用実行します。プログラムが異常終了した場合には、自動的にダンプファイルが作成されます。



1. 例外が起こったら、「・・・は動作を停止しました」というダイアログが出ます。



すぐに「プログラムを終了します」ボタンを押します。

ダンプファイルは、レジストリに設定したディレクトリ (今の例では C:¥CrashDumps)に作成されます。

🕌 CrashDumps					_ □	×
על-ם • 🖉 געל-ם	ディスク (C:) 🝷 Cras	hDumps	👻 🛃 🗸	irashDumpsの検索		<u>,</u>
整理 👻 ライブラリに追加	□▼ 共有 ▼	新しいフォルダー			:= 🔹 🔟 🤅)
🔶 お気に入れ	▲ 名前 ▲	•	更新日時	種類	サイズ	
く しんにくくう し ダウンロード 三 デスクトップ 「 最近表示した場所」	uniRTE.e	xe.1812.dmp	2009/11/27 15:10	DMP ファイル	97,659 KB	
浸 ライブラリ ■ ドキュメント ■ ピクチャ	-					



この方法では、自動的にダンプファイルが作成されますが、予めレジストリを設定しておく必要があ ります。

レジストリの変更をせずに、異常終了時にダンプファイルを作成するには、手作業になりますが、 ハングアップの時にダンプファイルを作成する方法を使うことができます(5.6「ハングアップ時にダ ンプファイルを作成させるには?(Vista、Windows 7、Windows Server 2008の場合)」を参照)

5.5. ハングアップ時にダンプファイルを作成させるには? (Windows XP および Windows Server 2003 の場合)

Windows XP では、ハングアップ時/エラー時にダンプファイルを作成させるには、ダンプを取ろうとするプロセスのプロセス ID を指定してワトソン博士を呼び出します。

このための前準備として、タスクマネージャでプロセス ID を見つけることができるようにしておく必要があります。 この方法は、5.7「タスクマネージャにプロセス ID を表示させるには?」を参照してください。

この設定をしてから、uniPaaSプログラムをテスト実行/運用開始します。

実行中にハングアップが発生した場合には、まず、ダンプをとるべきプロセスを特定する必要があります。プロ セスを特定する方法は、5.8「ダンプを取るプロセスのプロセス ID を特定するには?」を参照してください。 ダンプを取るべきプロセスのプロセス ID がわかったら、そのプロセス ID を指定して、ワトソン博士のコマンドに より作成します。

C:#WINDOWS¥system32¥cmd.exe



プロセス IDを指定して、ダンプファイルを作成させるには

- 1. コマンドプロンプト(DOS 窓)を開き ます。
- drwtsn32 -p (プロセス ID) と入力し、
 Enter キーを押します。
 右図は、プロセス ID = 2232 を指定した場合です。
- 3. 右図のようなダイアログが出るので、 しばらく待ちます。





 しばらくすると(1分以下)、「キャンセ ル」ボタンが「OK」ボタンに変わりま す。この後、OK ボタンを押します。 このとき、リッチクライアントの画面も 消えるはずです。



作成されたダンプファイルは、Windows XP の場合には

C:¥Documents and Settings¥All Users¥Application Data¥Microsoft¥Dr Watson

にあり、Windows Server 2003 の場合には

C:¥Documents and Settings¥Administrator¥Local Settings¥Application Data¥Microsoft¥Dr Watson

にあります。

C:¥Documents and Settings¥All Users¥Aj	opplication Data¥Microsoft¥Dr	Watson	
ファイル(E) 編集(E) 表示(V) お気に入り(<u>A</u>) ツー	-ル(T) ヘルプ(H)		
アドレス(D) 🗀 C:¥Documents and Settings¥All Users¥A	Application Data¥Microsoft¥Dr Watso	n	🖌 🄁 移動
7#11\$ ×	名前 サイズ サイズ	種類 更新日時	
	T45,481 KB	DMP ファイル 2010/08/10 144 テキスト ドキュメント 2010/08/10 144	7 17

5.6. ハングアップ時にダンプファイルを作成させるには? (Vista、Windows 7、Windows Server 2008 の場合)

Windows XP の場合と異なり、タスクマネージャにプロセス ID を表示させるための前準備は必要ありません。

uniPaaSプログラムをテスト実行/運用している途中で、ハングアップが起きた場合には、問題のプロセスを特定する必要があります。これは Windows XP の場合と同様、5.8 「ダンプを取るプロセスのプロセス ID を特定するには?」を参照してください。

ダンプを取るべきプロセスが特定できたら、次の手順でダンプファイルを作成します。

タスクマネージャで、ダンプを取るプロセスにカーソルを置きます。



 右クリックしてメニューを開き、「ダン プファイルの作成(C)」を選びます。



 ダンプを書き込んでいる旨のダイア ログが出て来ますので、しばらく 待ってください。(通常1分以内) 	ダンプのプロセス ・ プロセスをファイルに書き込んでいます。しばらくお待ちください…
	キャンセル
 しばらくすると、右図のような画面に なります。ファイル名を控えてから、 「OK」ボタンを押します。 	ダンプのプロセス Image: Comparison of the system of the syst

ダンプファイルは、上記のダイアログに示された場所にあります。





ここに説明した、タスクマネージャからダンプ ファイルを作成する方法は、異常終了時にダ ンプファイルを作成させる場合にも使うことが できます。

異常終了時には、右図のようなダイアログが 出ますので、このダイアログが出たら、タスク マネージャを開き、上記の手順でダンプファイ ルを作成できます。その後、「プログラムを終 了します」ボタンを押します。



この方法を使うと、毎回手作業でダンプファイルを作成する必要がありますが、5.4「異常終了時に ダンプファイルを作成させるには?(Vista、Windows 7、Windows Server 2008の場合)」に説明し たような、レジストリの変更を伴なう前準備は必要ありません。

5.7. タスクマネージャにプロセス ID を表示させるには?



タスクマネージャにプロセス ID を表示させるには (前準備として、一度だけ実行します)

1. タスクバーから、タスクマネージャを起動します。



2. 「プロセス」タブを開きます。

📇 Windows タスク マネ	< 一ジャ		
ファイル(E) オプション(O)	表示(⊻) シャットダウン(世)	ヘルプ(円)	
アプリケーション プロセス	パフォーマンス ネットワーク	ユーザー	
winlogon.exe w3dbsmgr.exe VMwareUser.exe VMwareTrayexe VMwareTrayexe VMwareService.exe task.mgr.exe System Idle Proce System svchost.exe svchost.exe svchost.exe svchost.exe svchost.exe svchost.exe spolsv.exe service.exe ServerHost_CS.exe PrinterService.exe	エリ・ユー・ロー・ 0 SYSTEM 00 magic 00 magic 00 SYSTEM 00 SYSTEM 00 SYSTEM 00 SYSTEM 00 LOCAL SERVICE 00 NETWORK SERV 00 SYSTEM 00 SYSTEM 00 Started <td>3,609 K 13,800 K 4,504 K 3,392 K 2,100 K 7,164 K 28 K 260 K 4,540 K 3,616 K 21,208 K 4,561 K 21,208 K 4,352 K 4,352 K 4,352 K 4,352 K 4,354 K 4,355 K 2,7216 K 2,7216 K</td> <td></td>	3,609 K 13,800 K 4,504 K 3,392 K 2,100 K 7,164 K 28 K 260 K 4,540 K 3,616 K 21,208 K 4,561 K 21,208 K 4,352 K 4,352 K 4,352 K 4,352 K 4,354 K 4,355 K 2,7216 K 2,7216 K	
プロセス: 34 CPU 倶	用率: 2% コミット チャー	ジ: 309MB / 990MB	

3. メニュー「表示 → 列の選択」を選びます。

77	·一ジャ			
2)	表示(⊻)	シャット	ダウン(<u>U</u>)	AN
:ス	最新の 更新の	2-4		
(列の選	択(<u>S</u>)		メモリ・
	SYSTEM magic		00 NN	3, 13

 「PID (プロセス ID) にチェックを入れ、「OK」を 押します。



5.「PID」という欄が追加されたことを確認します。

この後、タスクマネージャは閉じてかまいませ ん。

📕 Windows タスク 🗟	ネージャ			
ファイル(<u>E</u>) オブション(<u>O</u>) 表示(⊻) シャ:	ットダウン(山) ヘルプ(H)	
アプリケーション プロセン	、 パフォーマンス	ネットワーク ユーザー	-	
イメージ名	PID 1-t	げー名 C	メモリ使	^
winlogon.exe	428 S7ST	EM 00	3,608 K	
W3dbsmgr.exe V/Mwareliser.eve	2164 magic 1464 magic	. 00	13,800 K 4,504 K	
VMwareTray.exe	2596 magic	. 00	3,392 K	_
VMwareService.ex	e 160 S <mark>Y</mark> ST	EM 00	2,100 K	
taskingriexe System Idle Proce	1260 magic 0 SVST	: UU FM 98	7,180 K 28 K	
System	″ 4 S <mark>Y</mark> ST	EM 02	260 K	
svchost.exe	916 LOCA	L SERVICE 00	4,532 K	
svchostiexe	748 SVST	FM 00	3,010 K 21,220 K	
svchost.exe	684 NETV	ORK SERV 00	4,352 K	
svchost.exe	636 SYST	EM 00	5,140 K	
spoolsv.exe	356 SVST	EM 00	0,324 K 404 K	
services.exe	472 SVST	EM 00	3,456 K	
ServerHost_CS.ex	 1492 SYST 	EM 00	14,536 K	
PrinterService.exe	1088 5751	EM UU	27,216 K	
□ 全ユーザーのプロ・	2スを表示する(S)		プロセスの終了(E	
プロセス: 34 CPU	使用率:2%		48 / 990MB	
	100713-1-1-270			

タスクマネージャの設定は、レジストリに保存されるので、この操作は一度実行すれば、その後ずっと有効になります。

5.8. ダンプを取るプロセスのプロセス ID を特定するには?

ハングアップを起こしているプロセスを特定するには、タスクマネージャのプロセスの一覧から状況をみて判断 する必要があります。プロセスを特定したら、タスクマネージャの「PDI」欄からプロセス ID を知ることができます。

プロセスを特定するには、最初に「イメージ名」により、区別します。タスクマネージャの「イメージ名」欄には、プロセスの実行可能ファイル名が表示されます。uniPaaS製品のモジュールでは、次のような名前になっています。

製品	イメージ名
uniPaaS Studio	uniStudio.exe
uniPaaS 実行エンジン(Client、Enterprise Server、RichClient Server)	uniRTE.exe
リッチクライアントのクライアントモジュール	uniRC.exe
MRB	uniRQBroker.exe

例えば、Enterprise Server を使っているシステムで、サーバエンジンが動作を停止してしまっているようであれば、「イメージ名」が「uniRTE.exe」であるものを探します。

タスクマネージャに表示されているプロセスで、イメージ名が一致するものが一つしかなかった場合には、これ だけでタスクを特定できますが、イメージ名が一致するプロセスが複数ある場合には、その中から真犯人を区 別しなければなりません。これには確実な方法はなく、プロセスの挙動を見て推測する必要があります。

- MRB (uniRQBroker.exe)の場合、プロセスは二つありますが、区別は簡単です。一つは監視用のプロセスであり、メモリ使用量が少なく、もう一つは実際のブローカとしての仕事を行っているプロセスであり、メモリ使用量は多いです。ですので、メモリ使用量は多いです。ですので、メモリ使用量が大きいものを選べばよいということになります。
 Windows タスクマネージャント・ファイル(E) オブション(2015) また(E) ペルレブ(E)
 Windows タスクマネージャント・ファイル(E) オブション(E) オブレ(E) オブション(E) オブション(E) オブ・(E) オブション(E) オブション(E)
- 実行エンジン (uniRTE.exe)の場合には、
 「CPU」欄を見て推測します。ハングアップして
 いるプロセスは、CPUが 100%近くになってい
 るか(無限ループに陥っている場合)、あるい
 はいつまでたっても 0%のままか(待ち状態
 が永遠に続いている場合など)のいずれかに
 なっています。そのようなプロセスがあれば、
 それを選びます。
 右図の例では、プロセス ID が 2024 である
 uniRTE.exe が CPU 99% となっているので、こ
 れを選びます。

1 77	Windows タスク マネ 、 イル(E) オプション(<u>O</u>)	−ジャ 表示(<u>V</u>)	ヘルプ(円)			_ 🗆 🗙
7	プリケーション プロセス	パフォーマ	アンス│ネットワーク│ユ	ザ -		1
	イメージ名	PID	ユーザー名	CPU	メモリ使用…	▲
	wmiprvse.exe	2052	SYSTEM	00	5,388 K	
	winlogon.exe	356	SYSTEM	00	5.316 K	
	w3wp.exe	3616	NETWORK SERV	00	7.060 K	
	VMwareUser.exe	724	Administrator	00	6,400 K	
	VMwareTray.exe	644	Administrator	00	4,444 K	
	V/MwareService.exe	1376	SYSTEM	00	3,276 K	
1	uniRTE.exe	2024	Administrator	99	23,916 K	
	uniRTE.exe	1392	Administrator	00)	23,424 K	
	uniRQBroker.exe	3928	Administrator	- 80	9,500 K	
	uniRQBroker.exe	3912	Administrator	00	2,404 K	
	TscHelp.exe	980	Administrator	00	3,484 K	
	taskmerexe	1156	Administrator	00	6,760 K	
	System Idle Proce	0	SYSTEM	00	28 K	
	System	4	SYSTEM	00	268 K	
	svchost.exe	3504	SYSTEM	00	4,172 K	
	svchost.exe	1664	SYSTEM	00	4,640 K	
	svchost.exe	1460	SYSTEM	00	6,020 K	
	svchost.exe	1308	LOCAL SERVICE	00	1,392 K	
	svchostexe	1120	SYSTEM	00	2.408 K	_
	▼ 全ユーザーのプロセ	スを表示す	ta(S)		プロセスの終	7(E)
	セス: 41 CPU 修	• 東 用 率: 100		: 197MB	3 / 1254MB	

もし状況から判別するのが難しいようであれば、該当するプロセスのすべてについて、ダンプファイルを作成するのが一番確実です。

第6章 uniPaaS デバッガ

デバッガとは、アプリケーションのエラーを修正するために、開発者がアプリケーションを実行しながら、実行エンジンの状態やフローの流れなどを確認したり、手動で強制的に変更したりすることのできる、スタジオの機能です。uniPaaSでは、開発用のスタジオと、実行用のエンジンが分離されているため、デバッガが非常に使いやすく、かつ堅牢になっています。

本章では、以下のような uniPaaS でのデバッガの基本機能について解説します。

- ブレイクポイントを設定する。
- ブレイク時にタスクの項目の値を確認する。
- ブレイクポイントに条件を設定する。
- ウォッチリストを作成し、実行時に特定の項目の値を確認する。
- 実行時に項目の値を手動で変更する。
- スタジオ内で、実行フローを確認する。
- コールスタックを見る。
- 複数のコンテキストの実行状況を見る。



本章の内容については、以下の技術文書に基本的な事項についての説明がありますのでご参照く ださい。本書の内容は、主に、これらの補足説明をするものです。

デバッガー般	「リファレンスマニュアル」 アプリケーションのテスト → デバッガ
デバッガの使い方	「マスタリング Magic uniPaaS」 第 29 章 アプリケーションのデバッグ



重要: RichClient プログラムのデバッガについて:

- uniPaaS V1Plus Ver1.8 の RichClient プログラムは、デバッガに対応していません。このため、RichClient プログラムを実行中にデバッガを使って、ブレイクをかけたり項目をウォッチしたりすることができません。この制限は将来のバージョンで対応される予定です。
- アクティビティモニタは RichClient プログラムでも対応しています。このため、RichClient プロ グラムの実行の様子をアクティビティモニタで追っていくことは可能です。アクティビティモニタ はリモートデバッグでも利用することができます。(7.4「アクティビティモニタを表示するため だけに使うには?」を参照)

6.1. デバッガを開始する

デバッガを利用するには、Magic uniPaaS Studio が デバッグモード で動作している必要があります。





デバッグモードを解除するには・・・

デバッグモードを解除するには、同じく デバッグ(D) メニューから デバッグモード(D) を再度選択します。 このメニューは、選択するたびにデバッグモードがトグルします。

6.2. ブレイクポイント

ブレイクポイントは、プログラム中に開発者が設定するもので、プログラムが実行時にその位置に到達したら、 エンジンはプログラムの実行を一時的に中断(ブレイク)し、開発者がそのときのプログラムの状態を調査するこ とができるようになります。

6.2.1. ブレイクポイントを追加する

Magic uniPaaS でのブレイクポイントは、プログラムリポジトリ中の任意のタスクのデータビューエディタ、あるい はロジックエディタ上の行に設定することができます。ただし、コメント行には設定することはできません。





- 1. データビューエディタ、あるいはロジックエディタを開きます。
- 2. ブレイクポイントを設定したい行にカーソルを置きます。
- 3. デバッグ(D) メニューから、ブレイクポイント(T)を選びます。F9 キーで設定することもできます。
 - ▶ ブレイクポイントが設定された行には、左側に赤色の 🛑 印が表示されます。
 - > ブレイクポイントの設定は、デバッグ(D)→ ブレイクポイント(T) メニュー (あるいは F9 キー)によりト グルします。

右図は、ロジックエディタにブ レイクポイントが設定された例 です。

🕗 TravelAgency - uniPaaS Studio V1Plus									
ファイル(F) 編集(E) 表示(V) プロジェクト(P) タスク環境(K) オプション(O)	<u>テ</u> ゙バ ッ	グ(D) ツール(T) ヘルブ	°(Н)						
🎦 😂 🍃 🕽 🗗 🔳 🔯 💷 🔯 💷 🖉 🖉 🛃	▶ ≣	美行(R)	F7	6 🎾 😥					
Ø30 21 - 顧客注文集計	7	*ラウザで実行(W)	Ctrl+Shift+F7						
データビュー ロジック フォーム		「 ロジェクトの実行(J) 向正(P)	Ctrl+F/						
1 回 T=9279 P=前 2 項目更新 ∀=項目 0 総注文数 値	D 7	**/、**//* ŧ-ト* (D)		-					
3 日 FFU1→F* F=m 4 項目更新 V=項目 P 未処理注文数 値 5 項目更新 V=項目 0 処理済注文数 値	7	°レイク(B)	Alt+Ctrl+F7	-					
6 コール S=打2 なり 1 確容注文レコードのス=10 7 フォーム S=打2 なり 1 確容注文レコードのス=10 1 ロードのス=10 1 ロードのス=10	чііі А Чії Л	テリア (S) テリア イン(I)	F10 F11						
8 アウジョン E=式 8 Delay(5) 9 曰 T=ウスク S=役	7	547° P9F(O)	Shift+F11						
10 I7- ₩-營告 2 「総注文類JJ: %Str(総注對		121(1) 121(1)	Ctrl+F11						
	№ 1	ンジンをリセット(E)	Ctrl+Shift+F9						
	辞 り	モートエンジンに接続(C)							
ブレイクポイントの設定/解除を行います		モートエンジンを切断(H)							

右図は、データビューエディタ にブレイクポイントが設定され た例です。

🕗 Tra	velA	gency - uniF	PaaS	Studio V1Plus									X
ファイル(F) 編集(E) 表示(V) プロジェクト(P) タスク環境(K) オプション(O) デバッグ(D) ツール(T) ヘルプ(H)													
1	: 🚖) 🗐 🔮		🔯 II 🖉 🗐 d	- 😤 Sa	- 8	= L	l 🖪 🖬 🕯	h 💷 🦧	3 🛯 😫	1	2 📑	20 5
6 97	ク21	- 顧客注文	₩ 1										×
デー	タビュ	1 ロジッ:	2 3	フォーム									
	1	M=メインソース	7	顧客		んだっと	1						
	2	C= カラム	1	M.顧客コード	[30]	N=数値	5						
	8	C=1514	2	N. 顧客名	[31]	A≕文字	30						
	- 4	V=% #1	- 1	0.総注文数	[0]	N=数値	10	範囲:0	終10	代入0			
	-5	V-変数	2	P.未処理注交数		N=數値	10						
	6	V=変数	3	Q.処理済注文数		N=数値	10						
	7	D-安 壬		按行计计									



ブレイクポイントの設定は、プロジェクトファイルに記録されます。このため、プロジェクトを閉じた後に再度開いたときにも、前に設定されたブレイクポイントはそのまま有効になっています。

6.2.2. プログラムを実行する

ブレイクポイントの設定されているプログラムを実行すると、次のときにプログラ ムの実行がブレイク(一時中断)します。

- データビューエディタに設定されているブレイクポイントの場合には、ブレイクポイントの設定されている行で定義されている項目の値に変更があったとき。
- ロジックエディタにブレイクポイントが設定されている場合には、ブレイクポイントの設定されている行まで実行が進んだとき。

	継続(C)	F7
	ブラウザで実行(W)	Ctrl+Shift+F7
d,	プロジェクトの実行 (コ)	Ctrl+F7
	停止(P)	
	$\overline{\tau}^{\circ} \pi^{\circ} \eta \eta^{\circ} \overline{t} - \overline{t}^{\circ} (D)$	
II.	7° 1/10(B)	Alt+Ctrl+F7
Į.	λ797°(S)	F10
	λテップイン(I)	F11
þ	λテップアウト(O)	Shift+F11
0	ブレイクポイント(T)	F9
æ	ウォッチ(こ追加(A)	Ctrl+F11

ブレイクポイントから実行を再開するには、デバッグ(D)メニューから、次のよう なオプションを選択できます。

メニュー	+	意味
継続	F7	実行を継続する。
停止		プログラムを強制的に終了する。
ステップ	F10	現在の行を実行し、次の行で再度ブレイクする。
ステップイン	F11	ステップと同じだが、現在の行がコールコマンドやイベント実行コマ ンドの場合、あるいは開発者定義関数を含む場合には、それにより 呼ばれるプログラム、タスク、イベントハンドラ、関数などの最初の 行まで進んだところで再度ブレイクする。
ステップアウト	Shift+F11	現在実行中のタスク、ハンドラなどを最後まで実行し、呼出元に 戻った時点で再度ブレイクする。

6.2.3. 実行中に強制的にブレイクする

プログラムの実行に予想以上に時間がかかる場合とか、期待していた動作 でない場合には、強制的にブレイクをかけて、現在のプログラムの状態を調 査したい場合があります。

このような場合には、プログラム実行中に、デバッグ(D)メニューから、ブレイ ク(B)を選択します。その時点で強制的にブレイクが起こり、スタジオでは、 実行中のタスクが自動的に開いて、以下のように、現在実行中の行にカーソ ルが移動します。

- オンラインタスクでユーザのデータ入力待ちの場合には、データビュー エディタが開き、現在フォーカスのある項目に対応する C=カラム ある いは V=変数 などの行のカーソルが移動します。
- その他の場合(バッチタスクの実行中、あるいはオンラインタスクでもタ スク前・後処理やレコード前・後処理実行中の場合など)には、ロジック エディタ上で、現在実行中の行にカーソルが移動します。

	実行(R)	F7
	ブラウザで実行(W)	Ctrl+Shift+F7
	プロジェクトの実行(コ)	Ctrl+F7
-	停止(P)	
	$\overline{\tau}^{\circ} \mathcal{N}^{\circ} \eta \mathcal{I}^{\circ} \overline{t} - \overline{t}^{\circ} (D)$	
	ブレイク(B)	Alt+Ctrl+F7
	1717 (S)	F10
Ч	ステッフ [*] イン(I)	F11
dill.	ステップ [®] アウト(O)	Shift+F11
1	ブレイクポイント(T)	F9
æ	りオッチlこ追加(A)	Ctrl+F11
H	エンジンを地外(E)	Ctrl+Shift+F9
17	リモートエンジン/こ接続(C)	
đ.	リモートエンジンを切断(H)	

6.2.4. ブレイクポイントリポジトリ

設定したブレイクポイントは、ブレイクポイント リポジトリ で一覧表示できます。

ブレイクポイント リポジトリは、**表示(V)** メニューから、ブレイクポイント(B) を選 択すると表示されます。

ここには、設定されたブレイクポイントが一覧で表示されます。

7*	レイクホ [°] イント							×
X) 🕅 🕅	XII 🖿						
#	有効	タスク	パントドラ	行	条件	値	*	14
1	1	顧客注文集計	· レコード前	6	A=常時			_
2	1	- 顧客注文集計		5	D=データ更新	未処理注文数		
								-



有効欄は、ブレイクポイントの有効性を示すチェックボックスです。開発者は

チェックボックスをクリックすることにより、設定をトグルさせることができます。 有効 にチェックが入っていないブ レイクポイントは一時的に無効となったブレイクポイントで、設定された行まで実行が進んでもブレイクはかかり ません。

ブレイクポイントリポジトリで、右マウスクリックすると、コンテキストメニューが表示されます。

7*14	(クポ゜イント							
*	1 🕅 🕅	গ 🔄						
#	有効	<u> </u>	ルドラ	行	条件	値		· ·
1	v	顧客注文集計	니다.前	a 🛛	A=常時			
2	V	顧客注文集計		5	行削除(D)		F3	
					ソース(こ移動(G)		3
					特性(P)		Alt+Enter	

メニュー項目	意味
削除(D)	ブレイクポイントを削除します。
ソースに移動(G)	ブレイクポイントの設定されているタスクが開き、その行にカーソルが移動します。
特性(P)	ブレイクが起こる詳細な条件を設定することができます。詳しくは、後述の「条件付ブレイ クポイント」で説明します。

ブレイクポイントオプション

ブレイクポイントリポジトリにはツールバーがあり、次のような操作を行うことができま フット

7 61	(ク 木 [®] 1	0h		
1	Ø	Ŋ	Ø	1

アイコン	対応するメニュー	機能
*	削除(D)	現在カーソルのある行のブレイクポイントが削除されます。
XI		現在設定されているすべてのブレイクポイントが削除されます。
1		すべてのブレイクポイントが有効化されます。

31		すべてのブレイクポイントが無効化されます。
I	特性(P)	ブレイクが起こる詳細な条件を設定することができます。詳しくは、後 述の「条件付ブレイクポイント」で説明します。

条件付ブレイクポイント

ブレイクポイントリポジトリでは、各ブレイクポイントに対し、ブレイクする条件を設定することができます。

- ブレイクポイントリポジトリで、条件を 設定したいブレイクポイントにカーソ ルを合わせます。
- 右マウスボタンをクリックし、コンテキ ストメニューから 特性(P) を選びま す。

7°14	'ሳቱ° ብント						
1	1 9 X	🧌 💽					
#	有効	<u> </u>	ルント・ラ	行	条件	値	<u>^</u>
1	1	顧客注文集計	い下前	a	A=常用	i	
2	1	顧客注文集計		5	D	行削除(D)	F3
						ソースに移動(G)	
					C	特性(P)	Alt+Enter
					~	h	

 ブレイクポイント特性 ダイアログが 開きますので、ここでブレイクの条件を、次のオプションから選んで設定します。

ブレイク 値	意味	ダイアログイメージ
A=常時	(デフォルト)ブレイクポイントに達したら常にブ レイクします。	プレイクホ ペント特性 × プレイク: A=常時 C=前か D=条件 フ・レイクホ ペノトキセ D=条件 フ・レイクホ ペノトトニ達する度に処理が停止します.
C=カウント	回数 欄に設定された回数だけ、このブレイクポ イントを通過するたびにブレイクします。	ア [*] レイクポ [*] イント特性 ア [*] レイクポ [*] イント特性 ア [*] レイク: <u>D=ガウント</u> 回数 3 処理コマンドが指定された回数実行されると、 ア [*] レイ外**イントが有効になります。 OK(<u>0</u>) キャンセル(<u>C</u>)
N=条件	条件欄に式を設定し、ブレイクポイントに達した ときに、この式を評価した結果、真になった場合 にのみブレイクします。 条件欄に設定するには、プログラムリポジトリ で、ブレイクポイントの設定されているタスクを 開いておく必要があります。その状態で、条件 欄からズームして、式テーブルで設定します。	ア・レイクホ° インド特性 × ア・レイクホ° インド特性 × タート ト・ 条件 M > 5 この条件は、ア・レイクホ° インドに達する度にチェックされる論理値です。 OK(0) キャンセル(0)



条件は、ロジックビューで定義されたブレイクポイントに対してだけ設定することができます。データ ビューに設定されたブレイクポイントには、条件を設定することはできません。

6.3. 行の無効化

デバッグを行っている段階で、コールコマンドや項目更新コマンドなどを一時的に実行しないようにして、結果 がどう違ってくるかを確認したい場合があります。また、実行結果に影響を与えないと分かっているプログラム の実行を一時的に省略したいときがあります。

このような場合、今までは処理コマンドの「条件」欄を No に設定したり、あるいは「デバッグフラグ論理変数」の ようなものを定義しておいて、コマンドの「条件」欄にその変数を設定したりしていました。しかし、これはプログ ラムの変更を伴なう作業であり、デバッグが終了した後に設定を戻し忘れたりする可能性がありました。

uniPaaS V1Plus では、デバッガで実行中に、特定の処理コマンドを一時的に無効化する機能があります。この 機能はデバッグ途中でダイナミックに設定・解除することができるので、いちいちプログラムを変更する必要が なくなりました。



行を無効化するには

- プログラムリポジトリを開き、無 効化したい行にカーソルを置き ます。
- 右マウスクリックでコンテキスト メニューを開き、「行を無効化」 を選択します。

G) 927	21	- 雇	喀 注文集計	t							
5	ドータ	Ľ.	<u> </u>	ロジック	フォーム							
		1	🗆 T:	-929	P=前							
		2		項目更新	∀=項目	0	総注文数		値:	1	0	
		3	🗆 R:	ミレコートド	P=前							
		4		項目更新	∀=項目	Ρ	未処理注文数		値:	1	0	
		5		項目更新	V=項目	Q	処理済注文数		値:	1	0	
		6		3-10	S=#J2°929	1	顧客注文レコー	0	7* 1(7)	- L1	LE LE	条件
		7		77-6	□=出力	3	顧客注文	~	x -b(Z)		FD	L
		8		アクジョン	E=式	3	Delay(5)	1	行作成(R)		F4	L
		9	🗆 T:	=タスク	S=後				ヘッダ行作♬	苋(I)	Ctrl+H	L
		10		15-	₩警告	2	'総注文数は:'&S		行削除(D)		F3	L .
							(1	行を無効(۲(۲)	trl+Shift+D	
							```		特性(P)	_	Alt+Enter	I

無効化された行は、無効化されて いることがわかるように、薄い灰色 で表示されます。

-										
1	🕗 ዓスク	21	- 1	顧客注文集調	it .					
ſ	データ	ピュ	ı —	ロジック	フォーム					
		1 [	31	F=939	P=前					
		2		項目更新	∀=項目	0	総注文数	値:	1	0
		3 [	= F	ミーレコート	P=前					
		4		項目更新	V=項目	Р	未処理注文数	値:	1	0
		5		項目更新	V=項目	Q	処理済注文数	値:	1	0
		6	(	a-1	S=サフドタスク		顧客注文レコードのスキ			
		7		74-6	0-出力	3	顧客注文	7211:	1	Customer Orders
		8		アクション	E=式	3	Delay(5)			-
		8		フォーム アクジョン	<del>UF出力</del> E=式	3 3	観台/主义 Delay(5)	77111:	1	Customer_Urders



無効化は、メニュー「編集 → 行を無効化」を選択しても行 えます。

O Travel/	Ager	ncy - uniPaaS S	Studio V1Plus						
7711/(F)	編集	[(E) 表示(V)	7° ወジェクト(P)	タスク環	境(K) オプション(O)	デドバック	(D)	୬-ル(T) ∿ルプ(H)	
1 🖻 🕯		‡γ>セル(A)	Ctrl+F2		🗶 🌆 🔍 🔳	B 🔒		😭 💷 🔊	1
D 920 2	0	変更取消(U)	Alt+Back						
データヒ	∽	切り取り(T)	Ctrl+X						
	Þ	]t°-(C)	Ctrl+C						
	ĥ	貼り付け(P)	Ctrl+V		文数	値:	1	0	
	*	行作成(R)	F4		理注文数	値:	1	0	
	$\overline{\mathbf{v}}$	∧ッタ* 行作成(I)	Ctrl+H		齐注文数	値:	1	0	
6		(=#IIIA(D)	52		主文レコードのスキ	·[0_/*5%	-妇		솎
		行則味(D)	F3		注文	ファイル:	1	Customer_Orders	
		<u>すべて選択(S)</u>	Ctrl+A		/(5)				
	Æ	行を無効化(L)	Ctrl+Shift+		E文数は:'&Str(総注	表示:	B=†i	°ックス	



無効化された行を元に戻すには、無効化の場合と同様に、コンテキストメニューから「行を有効化」 を選択するか、メニュー「編集 → 行を有効化」を選択します。

# 6.4. ブレイクポイントと行の無効化についての注意

グレイクポイントやウォッチリストなどのデバッグ設定情報は、プロジェクトファイルには保存されません。プロ ジェクトの .EDP ファイルと同じディレクトリにある .OPT ファイルに保存されます。この .OPT ファイルは次のよ うな性質があります。

- プロジェクトに必須のものではありません。Studio がプロジェクトををオープンする際に見つからなければ、 Studio が自動的に作成します。
- ブレイクポイントやウォッチリストなど、デバッガの設定情報を保存します。
- .ECF ファイルやリポジトリ出力ファイルには、この情報は反映されません。このため、(1) ECF ファイルで 実行時には影響がありません、(2) リポジトリ出力・入力でプロジェクトを移行すると、デバッグ情報は消え ます。
- .OPT ファイルを削除すると、デバッグ設定情報はクリアされます。

ー方、行の無効化の設定情報は、プロジェクトファイル(各プログラムに対応した、Sources¥Prg_xxxx.xml ファイル)に記録されます。このため、

- 無効化されている状態で ECF ファイルを作成し実行すると、実行時にも無効化されたままとなります。
- プロジェクトをコピー (リポジトリ出力・入力)しても、無効化情報はそのまま移行されます。

デバッグが終了した時には、無効化されている行を元に戻すことを忘れないように注意してください。

### 6.5. 項目一覧

**項目** オプションでは、現在実行中のコンテキスト中の、項目の値を調べ、必要に応じて強制的に変更すること ができます。

### 6.5.1. 項目一覧の表示

**項目** 一覧は、表示(V) メニューから 項 目(V)を選択して表示させます。

- uniP	PaaS Studio V1Plu	JS	
表示	₹(V) プロジェクト(P	) オプション(O)	Ŧ
R	ታビゲータ(N)	Alt+F1	ł
	特性シート(P)	Alt+F2	ł
5	チェック結果(K)	Alt+F3	1
۲	ד)יר [°] אילאנ)	Alt+F12	1
¢₽	ペインの切替(W)	Ctrl+Tab	
	ዎクティヒ [、] ティモニタ(M)		1
	7*1/1/2# 121(B)		1
<b>6</b> 2	項目(∨)		1
25	3-#2990(S) 43		1
	実行コンテキスト(C)		1
66^	<u> </u>		

右図に示すように、現在有効な項目名、 型、データソース名、現在の値が表示 されます。

項目				x
& 13 🕲 🛡				
名前	型	ディータソース	値	*
顧客注文集計				
顧客コート	N=数值	顧客	1	
顧客名	A=文字	顧客	Ron Davis	
総注文数	N=数值	変数	0	
未処理注文数	N=数值	変数	0	
処理済注文数	N=数值	変数	0	
				_



項目一覧は、プログラムがブレイクしている状態でだけ有効です。

### 6.5.2. 項目一覧のオプション

項目一覧画面では、次のようなオプションが、右マウスボタンのコンテキストメニュー、あるいはツールバーから 操作することができます。

ツールバー			
	アイコン	メニュー	意味
項曰	4	ウォッチに追加	この変数を、ウォッチリスト に追加
😹 🗃 🕲 💻	660		します。ウォッチリスト については、
			6.6 ウォッチリスト を参照してくだ
			さい。
コンテキストメニュー	- <b>→</b> 2 ¹	ソースに移動	この変数が定義されている、デー
行削除(D) F3			タビューエディタの行に移動します。
ウォッチに追加(A) Ctrl+F11	<b>C</b>	データの設定	この項目の値を手動で変更します。
ソース(こ移動(G)	<u>کا</u> ا		
データの設定(S) F5			
データにNullを設定(N) Ctrl+U	NUU	データに Null を設定	この項目の値を、Null に設定しま
Expand the Data Plus	T		す。
Collapse the Data Left	L		1



データの設定(S) あるいは データに Null を設定(N) によって項目の値が変更された場合には、「新し い値によって影響する全ての式を再計算します か?」という確認ダイアログが表示されます。 「はい」と答えたら再計算ルールに従って再計算が 起こり、「いいえ」ならば再計算が起こりません。 「キャンセル」と答えると、変更した内容が元の値 に戻されます。



### 6.6. ウォッチリスト

**ウォッチリスト**は、項目一覧と同様、項目の現在の値を調査したり設定したりするものですが、項目一覧の場合、現在有効な項目がすべて表示されるのに対し、ウォッチリストは、開発者が指定した項目だけを表示するところが異なります。

#### 6.6.1. 項目をウォッチに追加する

ウォッチリストは、初期状態では空になっています。



 プログラムリポジトリのデータビューエディ タで項目定義をしている行にカーソルを置 き、右マウスボタンでコンテキストメニュー を出して、ウォッチに追加(A)を選びます。



● 項目一覧から、コンテキストメニューで ウォッチに追加(A)を選択します。



ウォッチリストでは、項目一覧と同じように、ソースに移動、データの設定、データに Nullを設定、などの操作を、 ツールバー、あるいはコンテキストメニューから行うことができます。

### 6.6.2. ウォッチリストの表示

ウォッチリストは、**表示(V)**メニューか ら **ウォッチ(H)**を選んで表示させます。

/ - uniPaaS Studio V1Plus								
:)	表示	ξ(V) ን° ፬୬° ェፇኑ(P)	オプション(O)	7*)				
	R	ታĽ*ታ*-9(N)	Alt+F1	ł				
	-	特性シート(P)	Alt+F2	h				
	<b>=</b>	チェック結果(K)	Alt+F3	II.				
	٩	∃メン™°イン(T)	Alt+F12	II.				
	₿	^° 心の切替(W)	Ctrl+Tab	I				
		ዎクティヒ [®] ティモニタ(M)						
	, M	ጋ [~] ሀብዕቱ° ብント(B)		II.				
	6.)	項目(V)		II.				
	<u>-</u>	]−ルスタック(S)		II.				
	-	実行コンテキスト(C)		Ш				
(	66	0799 (H)						

ウォッチリストには、項目一覧と同じよ うに、項目名、型、データソース名、お よび、現在の値が表示されます。

9797				x
× 歯 雪 ■				
名前	型	ディータソース	値	
顧客名	A=文字	顧客	Ron Davis	
総注文数	N=裝灯直	変数	0	
				-



項目一覧の場合と同様、これらの表示は、プログラムがブレイクしている場合にのみ有効です。
## 6.7. コールスタック

コールスタックは、メインプログラムから現在実行中のプログラムまでの、呼び出しの経路を表示します。

#### 6.7.1. コールスタックの表示

コールスタックは、表示(V)メニューから コールス タック(S)を選んで表示させます。

- uniF	PaaS Studio V1Pl	us - Executing	
) 表示	床(V) プロジェクト(P	) オプション(O)	Ŧ*
	ታビゲータ(N)	Alt+F1	H
	特性シート(P)	Alt+F2	h
=	チェック結果(K)	Alt+F3	
9	אלאנ (T)	Alt+F12	II.
¢⊟	ペインの切替(W)	Ctrl+Tab	II.
٩	ዎクティビティモニタ(M)		Ш
M	ブレイクポイント(B)		
64	項目(V)		II.
(=	]-#2997(S)	1 ²	
<b>60</b>	実行コンテキスト(C)	0	
60	ウォッチ(H)		

**コールスタック** ウィンドウには、モジュール名(プロ ジェクト名)、タスク名、ハンドラ名、およびそのハン ドラ中での行数が表示されます。

	1-1124	990				
l	🗖					
l		€9°a∽lk	<u> </u>	ルトッラ	行	~
l		TravelAgency	メインプログラム		0	
l		TravelAgency	航空会社	⊐ンՒ⊡∽⊮: Locations_PB	5	
		TravelAgency	航空会社	イベント:位置 <th>9</th> <th></th>	9	
	⊳	TravelAgency	航空会社位置	コントロール: Location_Code	5	
						~

ここでの表示は、プログラムがブレイクされて、実行が中断している状態でのみ有効です。

## 6.7.2. コールスタックのオプション

コールスタックウィンドウでは、コンテキストメ ニューから **ソースに移動(G)**を選ぶと、その タスクで現在実行中の行を、プログラムリポ ジトリ上で表示します。

3-112	990				
🛛					
	€୬°⊐−₩	<u> </u>	心ト"う	行	*
	TravelAgency	メインプログラム		0	
	TravelAgency	航空会社	コントロール: Locations_PB	5	
	TravelAgency	航空会社	へ"ント:位置 <th>9</th> <th></th>	9	
	TravelAgency	航空会社位置	コントロ "	Ĺ	Ŧ

3-112	990				x
🛛					
	€୬°⊐−₩	<u> </u>	ルト・ラ	行	
	TravelAgency	メインプログラム		0	
	TravelAgency	航空会社	コントロール: Locations_PB	5	
	TravelAgency	航空会社	へ"小:位置 <td>9</td> <td></td>	9	
⊳	TravelAgency	航空会社位置	리가마네: Location_Code	5	

オンラインプログラムで、ユーザからのデー タ入力待ちの状態のとき、あるいは、データ ビュー上でカラムや変数にブレイクポイント が設定されている場合にブレイクがかかった ときには、**ソースに移動(G)**を行うと、データ ビューエディタが開き、その項目を定義して いる行が表示されます。

ロジックエディタ設定されているブレイクポイ ントでブレイクしたときには、ソースの移動 (G)を行うと、ロジックエディタが開き、ブレイ クの起こった行が表示されます。

🕗 १२	13-	航空会社位	置						
デー	タビコ	ー ロジック	7 7	4-L					
	1	M=メインソース	5	航空会社位置		わデック	0		
	2	P=/\°ラメータ	1	M.航空会社コード_P	[27]	N=数值	5		
	3	P=/\°ラメータ	2	N.位置コード_P	[24]	N=数值	5		
	4	C= カラム	1	0.航空会社コード	[27]	N=数値	5	範囲:	4 新
⇔	5	C= カラム	1	P.位置コード	[24]	N=数值	5	範囲:	5 希

## 6.8. 実行コンテキスト

uniPaaSの実行エンジンはマルチスレッドで並行実行をサポートしています。

並行実行を使わないアプリケーションならば、実行コンテキストは一つなので、コンテキストの選択について気 を使う必要はありませんが、並行実行を使うアプリケーションでは、複数のコンテキストが同時に実行されてい ることがあります。

今まで見てきたコールスタック、項目一覧などは、各コンテキスト毎に独立して存在するものなので、正しい値を 調査するには、正しいコンテキストを選択することが必要です。

#### 6.8.1. 実行コンテキストの表示

実行コンテキストは、表示(V)メニューから、実行コンテキスト(C)を選択して表示させます。



実行コンテキスト ウィンドウには、コンテ キスト ID (Magic 実行エンジンが内部で 管理している番号)、コンテキスト名 (CtxSetName 関数で設定された名前)、 およびコンテキストの状態が表示されま す。

右図は、TravelAgency プロジェクトを実 行して、いくつかの並行実行プログラム を起動した状態で、ブレイクをかけたとき の 実行コンテキスト ウィンドウの例です。

*
A
-

この中で、# の欄に矢印 🎐 が表示されている行がありますが、これは現在選択されているコンテキストを表しています。項目一覧、ウォッチリスト、コールスタックなどの表示は、現在選択されているコンテキストのものが 表示されます。

#### 6.8.2. コンテキスト切替え

選択されているコンテキストを切り替える には、実行コンテキストウィンドウのコン テキストメニューあるいはツールバーから、 コンテキストの切り替え(S)を選択します。 コンテキストを切り替えると、項目一覧、 ウォッチリスト、コールスタックなどの表示 が更新され、新しく選択されたコンテキス トについての情報が表示されるようになり ます。

コンテキストを切り替えるには、 状態が停止済になっている必要があります。状態が停止中 になっている場合には、いったん、停止 済状態にあるコンテキストに対して、コン テキストの切り替え(S)をおこなってください。停止可能な状態であれば、停止中 が停止済に変更されます。

実行	17:	127421			
R	2				
#		ID	名前	状態	*
		17819021585568	Main	停止済	
5	⇒	26728532378352	26728532378352	停止済	
		35638043171136	AirlineLocations	停止済	
		44547553963920	44547553963920	停止済	
		53457064756704	<b>\$</b> 3457064756704		
		62366575549488	62366575549488	J)/f#XNO/U)督(S)	
				8	Ŧ



# 第7章 リモートデバッガ

リモートデバッガは、uniPaaS Studio 製品のデバッガ機能の一部として実装されていて、運用環境で、Client あるいは サーバ (Enterprise Server、RichClient Server) 製品を実行している環境でデバッグを行うための機能です。

開発が終了し運用開始した後に、問題が発生した場合、問題が運用環境でだけしか起こらないとか、問題の再 現が非常に難しい、ということがよくあります。このような場合には、運用環境でのデバッグが必要になってくる ことがあります。しかし、運用環境では Studio 製品が動いているわけではないので、前章で説明したような、 Studio 製品を使ってテスト実行しながらデバッグすることができません。

このような状況のために、uniPaaS では、通常の運用環境で運用をしながら、別 PC にある Studio を使ってデ バッグを行う、という方法をサポートしています。これを「リモートデバッガ」と呼びます。

本章では、リモートデバッガのしくみと使い方について説明します。



リモートデバッガについては、リファレンスマニュアルの「アプリケーションのテスト > デバッガ > リ モートデバッガ」の項も参照してください。

# 7.1. リモートデバッガのしくみはどうなっていますか?

前章で説明した通常のデバッガは、開発環境で Studio 製品を使って、プロジェクトを実行しているときの機能 でした。リモートデバッガについて説明する前に、通常のデバッグ環境における動作をより詳しく見ていきましょ う。

## 7.1.1. ローカルデバッグのしくみ

Studio を起動すると、開発環境としての Studio のプロセス (uniStudio.exe) の他に、背後では、テスト実行用に実行エンジン (uniRTE.exe) のプロセスが立ち上がっています。実行エンジンのプロセスは、開発状態では隠れていて表には現れませんが、F7 でテスト実行を行うときに表示されます。

テスト実行の際には、Studioと実行エンジンの間で TCP/IPによる通信セッションが作成され、デバッグの情報(ブレイクポイント、項目の値、アクティビティモニタの 出力、その他の制御信号)がそのセッションを通してやり とりされています。この時に使われる TCP/IP のポートを 「デバッグポート」と呼びます。

通常のデバッグでは、デバッグポートは uniPaaS が自動 的に決定するので、開発者はデバッグポートについて気 にする必要はありません。

#### 通常のデバッグ状態



──▶ デバッグ用TCP/IP セッション

なお、通常のデバッグの場合には、Studio はプロジェクトファイルを編集し、実行エンジンは、プロジェクトファイルを直接実行します。

#### 7.1.2. リモートデバッグのしくみ

リモートデバッグでは、このうちの実行エンジ ンが別マシン上で実行されるような形となり ます。ここでは、次のようになります。 実行環境では、実行エンジンだけが動作し ており、実行エンジンはプロジェクトのキャビ ネットファイル (ECF ファイル)を実行していま す。

実行エンジンはデバッグポートを開いており、 ECF ファイルの実行を行うと並行して、リ モートデバッグで Studio から接続されるのを 待機しています。

開発環境は、実行環境とは異なる PC 上で、 Studio を起動しています。ここには、ECF ファイルを作るもととなった プロジェクトファ イル (EDP ファイルおよびその Souces ディ レクトリ下のファイル) があります。



Studio はプロジェクトを開き、実行エンジンのデバッグポートに対して、デバッグ用の TCP/IP 接続を行ないま す。その後は、通常のデバッグの場合と同じように、Studio と実行エンジンとがデバッグ情報をやりとりしながら、 実行を進めていきます。

# 7.2. リモートデバッグにはどんな準備が必要ですか?

リモートデバッグを行うには、アプリケーションを実行している実行環境の他に、次の準備が必要です。

- 開発環境用の PC: Studio をインストールして利用するのに十分な性能を持った PC を1台用意します。 この PC は、実行環境の LAN に接続可能で、TCP/IP が正しく設定されている必要があります。
- Studio 製品: その PC に uniPaaS Studio をインストールし、ライセンスを正しく設定します。
- プロジェクト:実行している ECF ファイルのもととなったプロジェクトファイル (EDP ファイルおよびその Sources ディレクトリ下にある全 XML ファイル)を、開発環境 PC にコピーします。Studio で正常にプロジェ クトを開けることを確認しておいてください。このプロジェクトは、ソースファイルを参照するためだけのも のですので、開発環境 PC 上で実行できるように設定しておく必要はありません。
- ファイアウォールの設定: リモートデバッガは TCP/IP の通信を行うので、実行環境の側で、ファイアウォールのポートを開けておく必要があります。(7.3「リモートデバッグはどのように設定しますか?」参照)。
- デバッグユーザの ID とパスワード: セキュリティのために、リモートデバッグを行うためには、uniPaaS の セキュリティ機能を用いたユーザ認証が必要です(7.6「リモートデバッグのセキュリティは?」参照)。この ために、リモートデバッガ権利を持つユーザの ID とパスワードを知っている必要があります。

# 7.3. リモートデバッグはどのように設定しますか?

#### 7.3.1. デバッグポートの決定

最初に、デバッグ用の TCP/IP 通信に使う、デバッグポートを決めます。デバッグポートの番号について特に制限はありません。未使用のポートのなかから適当に選びます。

ここでは、説明の例のために、4040番を利用するようにします。

エンタープライズサーバやリッチクライアントサーバなどで、複数のインスタンスを起動する場合に は、個々のインスタンスごとに異なるデバッグポートを割り当てる必要があります。同一デバッグ ポートを指定して複数のインスタンスを起動すると、実行時のエラーやハングアップが起こることが あります。

#### 7.3.2. ファイアウォールの設定

リモートデバッグでは、TCP/IP を通して、実行エンジンと Studio とが通信するので、実行環境の側のファイア ウォールで、次のポートを開いておく必要があります。

ポート	説明	例
デバッグポート	未使用のポートのなかから適当に選びます。	4040
通信 GW ポート	「通信ゲートウェイ」テーブルの TCP/IP のエントリで、「パラ メータ」欄に定義されたポート番号の範囲。	1500-2000

「通信ゲートウェイ」テーブルは、実行エンジンをフォアグラウンドモードで開いて、メニュー 「オプション(O) → 設定 (S) → 通信ゲートウェイ (M)」 で開くことができます。



また、MAGIC.INI ファイルでは、[MAGIC_COMM] セクションの TCP/IP の行に設定があります。

 $[MAGIC_COMMS]$ NONE = 1, 0, NO Parameters needed TCP/IP = 2, 30, 1500-2000



#### 7.3.3. 実行エンジンの設定

実行エンジンでは、デバッグポートを開いて、Studioからの接続を待機するように、環境設定を行ないます。



デバッグ用に実行エンジンを設定するには

- 実行エンジンを起動して、メニュー「オプ ション (O) → 設定(S) → 動作環境 (E)」を 開きます。
- 9.「システム(S)」タブを開きます。
- 10.「リモートデバッガ」パラメータを Yes に、 「リモートデバッガの使用ポート」として、先 に決定したデバッグポート番号を指定しま す。この例では 4040 としています。

	×
定( <u>P)</u> 国別設定( <u>I</u> ) 外部参照(	( <u>E</u> ) ^{レッ} リケーションサーハヾ(!
ለ° 5メータ	
No	
Yes	
Yes	
0	
MGENT1P1	
744@Localhost	
Yes	
4040	•
OK	1 <b>キャンセル</b>
	定(P) 国際職業定(I) 外部参照(

この設定は、MAGIC.INIの [MAGIC_ENV] セクションでも設定できます。

[MAGIC_ENV] EnableRemoteDebugger = Y RemoteDebuggerPort = 4040

以上で、実行エンジン (サーバ)側の設定は終わりです。設定を有効にするには、実行エンジンの再起動が必要です。

#### 7.3.4. Studio からの接続

開発用 PC では、Studio から実行エンジンに接続を行ないます。



Studio から実行エンジンにリモート接続するには

- 1. 実行エンジンを起動した後に、開発用 PC で Studio を起動しま す。
- 2. 実行エンジンで実行しているアプリケーションの ECF ファイル のもととなった、プロジェクトファイル (EDP ファイル)を開きます。
- メニュー「デバッグ(D) → リモートエンジンに接続(C)」を選択します。



4. 「接続先の指定」ダイアログで、実行エンジンのあるサーバマシンのホスト名(あるいは IP アドレス)、デ バッグポート番号、および接続権限のあるユーザ ID とパスワードを指定します。

接続先の指定		
接続先の情報を入力してくた	iðu.	
詳細		
🔲 アウティビティモニタのみ		
ቱ⊼ŀ名/IPንト℃レス:	MyServer	
ホ°∽ト <del>番</del> 号:	40 40	
ב∽#°ID:	supervisor	
ስ°አዎ-ኑ*:	•••••	
	OK ( <u>0</u> )	)



接続時のユーザ ID/パスワードは、デバッグ時のセキュリティの確保のために指定する必要があ ります。デバッグ時のセキュリティについて特に設定していない場合には、supervisor として接続 すれば良いでしょう。デバッグ時のセキュリティについては、7.6「リモートデバッグのセキュリティ は?」を参照してください。

5. OK ボタンを押すと、実行エンジンに接続されます。

実行エンジンで実行しているアプリ ケーションの ECF ファイルと、Studio で開いているプロジェクトの対応が一 致していないと、右のようなエラー メッセージが出て、接続に失敗します。 プロジェクトと ECF ファイルの対応に ついて確認してから、再接続してくだ さい。



#### 7.3.5. デバッグ作業

Studioから実行エンジンに正しく接続できれば、デバッグを行うことができます。 リモートデバッグの場合には、ローカルのデバッグの場合と同様、ブレイクポイントの設定、ステップ実行、アク ティビティモニタの設定と表示、項目、コールスタック、実行コンテキストの表示などを行うことができます。 リモートデバッグの場合に利用できない機能も一部あります。例えば、リモートデバッグの場合、実行エンジン の停止を Studioから行うことはできません。また、リモートデバッグ中には、プロジェクトはすべて読み込み専 用となり、修正することはできません。

## 7.3.6. 実行エンジンの切断

デバッグ作業が終了して、これ以上接続しておく必要がなければ、Studio と実行エンジンの接続を切断します。



デバッガを実行エンジンから切断するには

メニュー「デバッグ(D) → リモートエンジンを切断(H)」を選択します。



2. 確認画面がでるので、「はい(Y)」を押します。これで、実行エン ジンとの接続が切断されます。

確認		<u> </u>
JT	ートエンジンを切断しま	ミすか?
	(\$\$\$)(Y)	いいえ(N)

# 7.4. アクティビティモニタを表示するためだけに使うには?

リモートデバッグ機能は、アクティビティモニタの表示のためだけに使う、特別なモードがあります。このモードでは、通常のデバッガとしての機能(ブレイクポイントの設定、項目値の確認や設定、ステップ実行など)は行うことができませが、アクティビティモニタをリアルタイムで見ることができます。また、アクティビティモニタの表示内容の設定も動的に変更することができます。



アクティビティモニタのみとしてリモート接続するには

- メニュー「デバッグ(D) → リモートエンジンに 接続(C)」で接続する際の「接続先の指定」ダ イアログで、「アクティビティモニタのみ」に チェックを入れます。 その他のパラメータは、通常のリモートデバッ グの場合と同じです。
- メニュー「表示(V) → アクティビティモニタ (M)」を選んで、アクティビティモニタ画面を開 きます。



Ctrl+Tab

アクティヒ、ティモニタ		
📄 🕨 🛯 🤌	袎	J₩ <u>1</u> 20
14:51:39.140	-	しいが読込
14:51:39.140	-	>>開始 ロード バッテ タスク ー 'e' 修正 モード
14:51:39.140	-	終了 奴切の-ト*
14:51:39.140	-	開始 IOファイルのオープン - ' (R)'
14:51:39.140	-	終了 I07ァ仙のオープン
14:51:39.140	-	開始 奴/前
14:51:39.140	-	7四 項目更新 'Date'
14:51:39.140	-	7四 項目更新 'Time'
14:51:39.140	-	7m 7ォーム O=出力 _ ^C:¥Documents and Setting:
14:51:39.140	-	終了 奴/前
14:51:39.140	-	lu-h*読込
14:51:39.140	-	開始 IOファルの加ーズー 'C:¥Documents and Settings¥A
14:51:39.140	-	終了 107ァイルのカロース*
14:51:39.140	-	開始 クローズ処理 バッチ タスク ー ゙e゙ 🛛 🛁
14:51:39.156	-	<<終了 ウローズ処理 タスク
14:51:39.156	-	- 開始 クローズ処理 バッチ タスク ー ゚メインプログラム (MyApp1 🎫
14:51:39.156	-	<<終了 ウローズ処理 ウスク
		▼
<		► H

□□□ へいの切替(W)

7071L*71E_9(M)
 7*10101

# 7.5. アクティビティモニタの内容を変更するには?

リモートデバッガで接続すると、実行エンジンの実行のようすをアクティビティモニタに表示させることができますが、ここに出力する内容の詳細を、ロギングダイアログで動的に変更することができます。



アクティビティモニタに出力する内容を変更するには

 メニュー「オプション (0) → 設定 (S) → ロギ ング」を選び、ロギング ダイアログを開きます。



ロギングダイアログで、設定を変更します。
 設定した内容は、即座に反映されます。





ロギング ダイアログの設定については、2.5 「ロギングダイアログでフィルタを設定するには?」あ るいは、リファレンスマニュアルの 「設定 → ロギング → [設定] タブ、および [DBMS] タブ」を参照 してください。

# 7.6. リモートデバッグのセキュリティは?

デバッガでは、実行中のアプリケーションについてほとんど全ての情報を取得することができます。例えば、プログラムのデータビューやロジックはもちろん、現在実行中のプログラムの項目の値やコールスタックなども見ることができます。また、項目の値やプログラムのコントロールもステップ実行により1ステップづつ、手作業で強制的に変更することも可能です。

このため、デバッガは大きなセキュリティホールになる可能性があります。これを防ぐには、デバッガのリモート 接続についても、誰でも自由にできてしまうのは不適当であり、デバッガの接続のための権限設定を行って、一 定の認証にパスした人だけが接続できるようにしておくべきです。

uniPaaSのリモートデバッガでは、セキュリティを確保するために、プロジェクトの「アプリケーション特性」で「リモートデバッガ権利」を設定できるようになっています。

リモートデバッガでのセキュリティの設定の設定にあたり、リモードデバッガ権利キーと、その権利を有するユー ザを決定しておく必要があります。

- リモートデバッガ権利のキーを、半角英数字10文字以内で決定します。ここでは例として、 「DEBUGCON」というキーを使いますが、実際にはなるべく推測しにくいキーを使ってください。
- リモートデバッガ権利を有するユーザの ID、パスワードを決定します。ここでは例として、ID は 「debugusr」、パスワードは「debug」とします。実際にはなるべく推測しにくい ID/パスワードを使ってください。

セキュリティの設定は、次の3ステップで行ないます。

- リモートデバッガ権利をプロジェクトに設定します。(開発環境)
- 実行環境で、リモートデバッグ用のユーザを登録します。(実行環境)
- リモートデバッグ接続時に、ユーザ ID/パスワードを指定します。(開発環境)

それぞれについて、以下に説明します。

#### 7.6.1. リモートデバッガ権利の設定

最初に、リモートデバッガの権利を定義し、プロジェクトに設定します。これは Studio で行ないます。



- 開発環境で Studio を起動します。
- プロジェクトの 権利リポジト リを開き、リモートデバッグ用 の権利を定義します。

0	権利リポジトリ				×
Ц.	名前	<b>‡</b> ~	公開	フォルジ公開名	*
	「デバッガ接続権利	DEBUGCON	No	$\mathbf{c}$	
					-
L					

- アプリケーション特性を開き、 「セキュリティ」タブの「リモー トデバッガ権利」からズーム します。「利用可能権利ー 覧」が表示されます。
- 一覧の中から、上記ステップ 2で作成したリモートデバッ ガ権利を選択します。
- アプリケーション特性を閉じ、 キャビネットファイル (ECF ファイル)を再作成します。
- 6. 作成した ECF ファイルを、実 行環境にコピーします。

77*リケーション特性	×
スタートアップ (S) 外部参照ファル(E) セキュリティ(B) セキュリティキー	利用可能權利一覧
このブロジェクトがアブリケーションのメインとなる場合に使用 権利キーを指定します、コンボーネントとしてオーブンされる場合 は無効になります。	表示 同一77°リケーション
₽ <b>フ</b> °ሣケーションፆウセスキー:	1 デバッガ接続権利
公開権利牛:	
スール°権利牛-:	
リモートティルショカ 権利: デバッガ 接続権利	
PPD	ixon

#### 7.6.2. リモートデバッグ用のユーザの定義

次に、リモートデバッグ時にログインするためのユーザを、実行環境で定義します。このユーザには、リモートデ バッガ権利を割り当てます。

実行環境で、リモートデバッグ用のユーザを登録するには

- 実行エンジンをフォアグランドモードで起 動します。
- 「オプション (O) → 設定(S) → ユーザ
   ID (U) 」を選んで、ユーザ ID テーブルを 開きます。
- 3. 新規ユーザを作成し、ユーザ ID (debugusr)とパスワード (debug)を設定 します。
- 4. 「権利」欄からズームして、1 行作成しま す。
- 5. リモートデバッガ権利のキー (DEBUGCON)を割り当てます。

🕗 ユーザID					x
# 2~ザID 名 1 SUPERVISOR SU	前 パ; JPERVISOR	አዎ-ኑ	権利	ታ°⊮-7° የ	0
2 debugusr	*	****	**		0
	権利:				X
	- プロジェクト名 MvApp1			¢ם°7	՝ _≖ ウト(P)
	I T T PEBUGCON	名前 デバッ:	ガ接続権利		
					-
				OK ( <u>0</u> )	キャンセル( <u>C</u> )

## 7.6.3. リモートデバッガの接続

以上の準備ができたら、Studioから実行エンジンにリモート接続します。



接続の方法は、7.3.4「Studio からの接続」と同じですが、「接続先の指定」ダイアログで、ユーザIDとパスワードとして、デフォルトの supervisor ではなく、先に定義したデバッグ用のユーザIDとパスワード(この例では debugusr/debug)を指定するところが異なります。

接続	先の指定		×
1	接続先の情報を入力してくだ		
	詳細		
	📃 アウティビティモニタのみ		
	标补名/IPንドレス:	MyServer	
	ホ°∽ト番号:	4040	
6	2-#°ID:	debugusr	
	ለ°አ9-ኑ°:		
		OK( <u>0</u> ) [ ‡ _* )/b	↓( <u>c</u> )

# 第8章 DBMS のユーティリティ

本章では、DBMS が提供しているユーティリティ群のうち、uniPaaS アプリケーションのデバッグの段階で便利に 使えるツールを紹介します。

ここでは、uniPaaS アプリケーションのデバッグの段階において有用と思われるツールの機能にフォーカスして 説明します。それぞれのツールの機能としては、本章で紹介する以上に、非常に豊富な機能を持っているもの も多いのですが、本章では割愛します。また、各ツールの具体的な利用方法の説明も省略しています。これら について、詳しくは、各 DBMS ユーティリティのマニュアルなどを参照してください。

# 8.1. Pervasive PSQL のツール

#### 8.1.1. Pervasive Control Center

Pervasive Control Center は Pervasive PSQL (以下、Pervasive と略称)のエンジン、クライアント、データベース 情報などの確認・設定のみならず、添付技術文書やログの確認など、Pervasive に関する作業の中心となる ユーティリティです。



ここでトラブルシューティング時に良く使うのは、次の二つです:

 ローカルエンジンの構成: この PC 上で実行されている Pervasive の DBMS エンジンに関するパラメータ の設定。

> 「エンジン」の下にある、PC 名 (上の 例では MyServer) のノードから、コ ンテキストメニューで「プロパティー」 を選択するか、あるいは「ようこそ」 画面の「ローカルエンジンの構成」 をクリックすると、構成画面が表示さ れます。



 MicroKernel ルータの構成:他の PC にデータベースエンジンがある 場合、データベースエンジンに接続 する際の、クライアント側のパラメー タの設定。「MicroKernel ルータ」ノー ドからコンテキストメニューで「プロ パティー」を選択するか、あるいは 「ようこそ」画面の「MicroKernel ルー タの構成」をクリックすると、構成画 面が表示されます。



## 8.1.2. Magic 実行時の標準的な設定

Pervasive では、uniPaaSの実行にあたって、安全性を高めてトラブルを極力回避するという意味で、一般に推奨している標準的な設定があります。Pervasive のテーブルのアクセスに問題があると思われる状況では、 Pervasive の Control Center において以下のような設定をしてみてください。これは各サーバおよびクライアントにおいて設定することになります。

カテゴリ	パラメータ	標準設定	備考
通信プロトコル	サポートプロトコル	不要なものは削除	通常は TCP/IP のみでよい
	自動再接続の有効化	オン	
	自動再接続タイムアウト	1800 秒(30 分)な どの大き目の値	
データ整合性	トランザクションー貫性保守	オン	Magic では必須の設定
メモリの使用	非アクティブ時、最小の状態に戻す	オン	長時間未使用時に応答なしと なる場合
パフォーマンス チューニング	セグメントサイズを2GB に制限	オフ	ファイルを分割したくない場合

#### ローカルエンジンの構成

#### Microkernel ルーターの構成

カテゴリ	パラメータ	標準設定	備考
通信プロトコル	サポートプロトコル	不要なものは削除	通常は TCP/IP のみでよい
	自動再接続の有効化	オン	
パフォーマンス チューニング	キャッシュエンジンの使用	オフ	更新処理が多い場合

## 8.1.3. ログ

Pervasive のレベル (DBMS エンジンおよびクライアントモジュール)に重要なエラーが発生した場合には、ログファイルに記録されます。

このログファイルは pvsw.log という名前で、Pervasive のバージョンやエディション、Windows のバージョンなど により格納されている場所が異なりますが、Pervasive Control Center のメニュー「ヘルプ(H)  $\rightarrow$  PVSW ログ」 を選んで表示させることができます。



## 8.1.4. トレース

uniPaaSのPervasive ゲートウェイは、Pervasiveのクライアントライブラリに対して、Pervasive が提供する API を介してアクセスしています。uniPaaSのプログラム実行中に、Pervasive テーブルへのアクセス時に問題が起こる現象が見られたら、詳細調査のために、この API レベルでのトレースを確認してみたくなることがあります。

Pervasive では API レベルのトレース機能を提供しており、これを使って API のアクセスの様子を確認することができます。



Pervasive のトレースを有効にするには

- Pervasive のデータベースエンジンが 実行されている PC 上 (DBMS サー バ上)で、Pervasive Control Center を開きます。
- 3. 「ローカルエンジンの構成」画面を開 きます。
- 「デバッグ」画面で、「トレースオペレーションの実行」をオンにします。トレースするオペレーションはすべてオンにしておいてください。他の設定はデフォルトのままで構いません。



Pervasive の「開発者用ドキュメント → PSQL Programmer's Guide → Btrieve アプリケーションのデバッグ」に、 「トレース ファイル」のサンプルがいくつかあり、その読み方も説明されています。 API のインターフェースはバイナリデータで行われており、またトレース結果もかなり詳細なものになるため、ト レースをとってのデバッグ調査は、かなり現象が絞り込めてきてからのことになると思われます。

#### 8.1.5. Butil

Butil は、Pervasive が提供するコマンドラインツールで、次のような幅広い機能を持っています。

- サーバーのバックアップに使用する Continuous オペレーションの開始と終了。
- 最後のバックアップからシステム エラーが発生するまでの間に行ったファイルへの変更の回復。
- ASCII 形式、シーケンシャル形式、および SDF 形式のデータのインポートとエクスポート。
- データファイル間のデータのコピー。
- MKDE バージョン情報の取得。

トラブルシューティング時には、次のような使い方があります。

#### Pervasive 製品バージョンの確認

引数として -ver とすると、Pervasive ソフトウェアモジュールのバージョンについての情報を表示します。

C:¥>butil -ver Btrieve Maintenance ユーティリティ 10.30.017.000 Copyright (C) Pervasive Software Inc. 2009 All Rights Reserved. BUTIL-33: Btrieve リクエスターのハー・ションは 10.30 です。 BUTIL-138: Btrieve のハー・ションは 10.30 で、Workgroup Engine 版です。 コマント が完了しました。

#### ファイルのヘッダ情報の確認

Pervasive テーブルへのアクセス時に、問題が起こるファイルと起こらないファイルがある場合、Butil -stat にお いてヘッダの部分の情報に差異がないかを確認することで、原因を見つけられることがあります。以下に実行 例を示します。

C:¥>butil -stat "C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥TREECTLTAB1" Btrieve Maintenance 1-74974 10.30.017.000 Copyright (C) Pervasive Software Inc. 2009 All Rights Reserved. File Statistics for C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥TREECTLTAB1 File Version = 9.50 Page Size = 4096 Page Preallocation = No Key Only = No Extended = No

Total Number of Records = 10			
Record Length = 66			
Record Compression = No			
Page Compression = No			
Variable Records = No			
Available Linked Duplicate Keys = 0			
Balanced Key = No			
Log Key = SYSKEY			
System Data = Yes			
SYSKEY Status = Present			
Total Number of Keys = 1			
Total Number of Segments = 1			
Key Position Type	Null Values*	ACS	
Segment Length	Flags Unio	que Values	
0 1 1 8 Float	M	0	



Pervasive のファイルを比較する場合、Butil -stat で比較する他に、エクスプローラのファイルのプロパティの情報(「詳細設定」ボタン押下時に表示される内容も含め)も比較してください。

#### 8.1.6. Function Executor

Pervasive テーブルに対して、レコードの更新・削除等を行っても、思った通りにならない場合、Pervasive が提供する Function Executorを使って、Pervasive の API レベルで直接オペレーションを行って見て、uniPaaS の側の問題なのかの、切り分けを行うことができます。

Pervasive Function E:	xecutor – [c:¥pro	ogram files¥u	ınipaas¥st	tudio v1p	lus¥projects	¥myapp14	
🗞 ファイル(E) Get(G) Ge	et Ext(∑) Step( <u>S</u> )	Updates( <u>U</u> )	表示(⊻)	ツール(T)	ウィンドウ(₩)	ヘルプ(日)	_ 8 ×
B C C 🖉 🖆	2 👩 🔇						
データ バッファー				C St	en (物理的)(S)	O Get (i	命理的ない
0000 42 00 00 10 01	00 00 00 00 00	Β				0.000.0	
000A 00 02 00 08 00	00 01 00 08 00			412 )		n/	
0014 02 01 00 00 00	00 02 00 00 00				· / ·- /-	70	
001E 00 00 00 00 00 00	00 00 00 00 00					. 🔻	
0028 00 00 00 00 00 00	00 00 00 00 00			-=-0	2該定 & 号(N) 0	J 🗆 ±-	
0032 00 00 00 00 00 00	00 00 00 00 00				e ologia –		170
003C 00 00 00 00 00	00 00 00 00 00			「繰り返		ック( <u>L</u> ) - ナN	<u>с ш</u> _
0046 00 00 00 00 00 00	00 00 00 00 00 00		•••	.   🛄		ゆェイト	○∓
キー バッファー							
0000 00 00 00 00 00	00 00 00 00 00			]   オペ =	コード( <u>O</u> ): 15	😵	実行
000A 00 00 00 00 00	00 00 00 00 00			STAT	. (15)		-
0014 00 00 00 00 00 00	00 00 00 00 00			, データ	馬 32		
001E 00 00 00 00 00	00 00 00 00 00				26 105		
0028 00 00 00 00 00 00	00 00 00 00 00						
0032 00 00 00 00 00	<u>nn nn nn nn nn</u>			<u>ا ا</u>			
			0		<u> </u>		
					171		11.

例えば、Pervasive のテーブルをオープンし、先頭のレコードを取得し、そのレコードを削除する、という一連の オペレーションがうまくいかない場合、Function Executor を使って、次のような操作を行ないます。

- 1. メニュー「ファイル → 開く」で該当ファイルをオープン
- 2. オペコードを 12 (GET FIRST)にして、実行。(最初のレコードに位置づけされる)
- 3. オペコードを 4 (DELETE) にして、実行(位置付けしているレコードが削除される)

Function Executor を使って、Pervasive ファイルの バージョンを調べることもできます。ファイルをオー プンしてから、メニュー「表示 → ファイル統計情 報」を選ぶと、ファイルについての情報が表示され ますが、その中の「ファイル バージョン」にバージョ ンが表示されます。

右図の例では、ファイルバージョンは 4.0 です。

D:¥tmp¥DMFIL001.DAT			_ 🗆 🗙
ファイル サイズ	8,192	ページサイズ (バイト)	2048
(ファイル バージョン	4.0	レコード数	1
レコード長	10	未使用のプリアロケート ページ	0
キー数 / セグメント数	1/1	未使用のリンク重複ポインター	0
オープン モード	ノーマル		
750			
レコード圧縮		空きスペース	5%
ページ圧縮		キーオンリー	
可変長レコード	<b>V</b>	インデックス バランス	
ブランクトランケーション		ページ プリアロケーション	
VAT			
D = Dups M = Mod	R = Rep Dup	s A∕L=Null <=Desc I=Casel	ins
キー番号 セグメント ユニーク値 (	立置長さ	属性データ型	
	1 4	M Integer	
<b>a !</b>			

#### 8.1.7. Pervasive System Analyzer

Pervasive System Analyzerとは、インストールされている Pervasive 製品について、以下のような一連のテスト を行い、レポートするツールです。

- Pervasive のクライアントとデータベースエンジン間のネットワーク接続をテストする。
- PervasiveのAPIを通してデータベース処理を行い、正しく機能しているかをテストする。
- インストールされている Pervasive コンポーネントの情報を表示する。

Pervasive System Analyzer は、ウィザード形式でテスト項目を選択していき、最後にテキストファイルでレポート を保存します。



Pervasive の DBMS エンジンやネットワーク接続の動作異常が疑われる場合には、このテストを行ってエラー等が発生していないかをレポートで確認してください。

#### 8.1.8. License Administrator

Pervasive のテーブルを使っているとき、「ファイルマネージャの異常 #161 エラー」が発生した場合には、 Pervasive のライセンスが不足していたり期限切れになっていることを意味します。 このような場合、Pervasive に付属の「License Administrator」を使用すれば、有効なライセンスが存在するかを ご確認できます。

6	) Pervasive So	oftware Lice	ense Administra	tor - MYSERVER								
	サーバー名:	MYSERVER	२		_	接続	参照( <u>B</u> )	7				
	+:						アクティブ化( <u>A</u>					
i i	ライセンス情報:				Luca	1	1 - 4				(	
			製品キー		状態	<u>  ブラットフ.</u>	<u>   ライセンス タイ</u>	<u>ਗ਼   ユー !</u>	f	有効期限	<u> </u>	アプリケ
	Pervasive PSQL	10 Workgro.	. 適用外		アクティブ	Win32	一時		5	2010/10/28	2	0
	Pervasive PSQL	10 Workgro	. 適用外		アクティブ	Win32	期限なし		5	無期限	2	0
F	▼ 期限切れを非	表示									非アク	ティブ化(D) 🌔
-												
								リフレッシュ(E)		ヘルプ		閉じる
									_			

#### 8.1.9. Pervasive Software Monitor

Pervasive Software Monitor は、Pervasive のデータベースエンジン (MicroKernel) の活動の状況を表示する ユーティリティです。このユーティリティで次のような情報を確認することができます。

- 接続しているユーザ(クライアント)の一覧、および、各接続ごとの情報。
- 現在オープンされているファイルの一覧、および、各ファイルごとの情報。
- データベースエンジンのリソース使用状況の統計。
- クライアントとの接続の通信統計情報。

これらの情報は、例えば、次のような状況で確認のために使うことができます。

- レコードロックの解除待ちが発生する場合、どのユーザが使っているのか?
- ライセンスエラーが発生する場合、誰がどこから接続しているのか?
- 通信障害が起こるときに、通信状況がどうなっているのか?

次の図は、Pervasive Software Monitor ユーティリティの画面例です。ここでは、アクティブファイルと、アクティブユーザについての情報が表示されています。



## 8.1.10. 再インストール

Pervasive が正しく動作しない場合、いろいろと試しても正常に戻らない場合には、最後の手段として、 Pervasive の再インストールを行うことがあります。

この場合、アンインストールしても、レジストリに情報が残っていて、設定がそのまま残っていると、再インストールした後にもそれが有効になって、本当の意味で「クリーンな環境に再インストール」とはなっていないこともあります。このため、Pervasiveの再インストール時にはレジストリもクリアするようにしてみてください。

具体的には、次のような手順を行ってください。



Pervasive をクリーンに再インストールするには

- 4. Pervasive のアンインストール
- 5. マシンの再起動
- 6. PSA(Pervasive System Analyzer)を起動して「コンポーネントまたはアーカイブを削除する」を実行 (Pervasive PSQL 10 では不要)。
- 7. マシンの再起動
- 8. PSA のアンインストール
- 9. マシンの再起動
- 10. 以下のフォルダの削除
  - C:¥PVSW
  - C:¥BTI
  - C:¥Program Files¥BTI
- 11. 以下のファイルの削除(特にパスの通っているフォルダにないかを確認)
  - wbtrv32.dll

第8章 DBMSのユーティリティ

- w3btrv7.dll
- 12. Windows のインストールフォルダ¥pvsw.log
- 13. レジストリエディタの起動(コマンドプロンプトから「regedit」を実行)
- 14. レジストリエディタの「レジストリ/レジストリファイルの書き出し」を実行(バックアップ用)
- 15. 以下のレジストリの削除
  - HKEY_LOCAL_MACHINE/SOFTWARE/Pervasive Software
    - HKEY_LOCAL_MACHINE/SOFTWARE/Btrieve Technologies
- 16. Pervasive の再インストール

#### 8.1.11. ファイルの互換性

Pervasive のファイルには、フォーマットにバージョンがあり、4.x ~ 10.x まで数種類があります。このうち 5.x お よびそれ以前のバージョンはトランザクション処理などが異なり、最新の Pervasive エンジン (9.x 以降)では読み 込み専用としてのみアクセスすることができます。もし、4.x、5.x のバージョンのファイルを、9.x 以降のエンジン を使ってアクセスし、書き込み (挿入、修正、削除)を行おうとすると、ステータス #46 のエラーとなります。 このため、古いバージョンのエンジンで作成したファイルを新しいバージョンのエンジンでアクセスするような状 況では、次の点に注意してください。

- 1. システムの移行などで、古いエンジンを使って新たにファイルを作成することがない場合には、BUTIL などを使って、ファイルを新しいバージョンで再作成してください。
- 古いエンジンを使うシステムと新しいエンジンを使うシステムとが共存し、ファイルコピーなどを行ってデータ共有を行うような構成の場合には、古いエンジンの「作成ファイルのバージョン」を 6.x 以上に設定してください。Control Center で設定 → サーバ →ファイル互換性 →作成ファイルのバージョンで設定します。(右図は Pervasive.SQL 2000i での設定です)





Pervasive ファイルのバージョンは、BUTIL や Function Executor を使って調べることができます。 8.1.5「Butil」および 8.1.6「Function Executor」を参照してください。

## 8.2. MS SQL Server のツール

#### 8.2.1. SQL Server Management Studio

SQL Server Manageent Studio は、SQL Server を管理するための基本的なツールです。これ自身にはトラブル シューティングの機能はありませんが、データベースの構成や設定、テーブルに関する情報等を細かく確認・設 定することができますので、運用時の管理目的だけでなく、テスト・デバッグ時にも必須のツールです。



#### 例1: 範囲付の結果が予想と異なる

UniPaaSでMS SQL Server のテーブルをアクセスする場合、範囲付や位置づけを行っているが、予期したのと は異なるレコードが検索されることがあります。このような場合には、データベースの「照合順序」が影響してい る可能性がありますので、SQL Server Management Studio から照合順序を確認してください。



 「オプション」を選択します。「照 合順序」で、設定を確認できま す。

j データベースのプロパティ - Mi	AGIC			
ページの選択	🖾 スクリプト 🔸 📑 ヘルプ			
☆ ファイル <del>~ ファイル ジ</del> ループ	照合順序( <u>C</u> ):	Japanese_CI_AS		
「オノション	復旧モデル(M):	単純		*
☆ 拡張プロパティ	互換性レベル(止):	SQL Server 2005	(90)	~
	その他のオプション(型):			
	□ カーソル			^
	コミットでカーソルを閉じる 既定のカーソル		False GLOBAL	_
	□ その他			
	ANSI NULL 既定値		False	
	ANSI NULL 有効		False	



MS SQL Server の照合順序の意味については、Microsoft 社の Web サイト http://msdn.microsoft.com/ja-jp/library/ms143515.aspx などを参照してください。

#### 例2: テーブルのプロパティを確認する

uniPaaSのデータベース特性の「開発モードでのテーブル変換」がオフの場合、uniPaaSのテーブルリポジトリを変更すると、uniPaaSでの定義とSQL Server上での実際のテーブルの定義とが異なることで問題が発生することもあります。

例えば、uniPaaSでカラムを追加した場合、新し いカラムは SQL Server のテーブルには存在し ていないので、「列名: (列名)が無効です」とい うエラーが出ます。

このような場合には、SQL Server Management Studioを使って、SQL Server 上でのテーブル定 義 (列やキーのプロパティ)を確認してください。

また、uniPaaSの日付型は SQL Server 上では datetime 型にマッピングされますが、SQL Server の datetime 型では「0000/00/00」という 日付は不正であり、uniPaaS から入力しようとす るとエラーとなります(右図)。このような場合にも カラムの特性を確認して、CHAR 型でテーブル を作成するか、あるいは 0000/00/00 という日 付が入らないようにプログラム上改修を行ってく ださい。

BIJ-	
AREYOMI : 列名 'Phone2' が無効です。	
	OK

DBI7-	
○説明: キャストした文字コードが正しくありません。	
	(OK)

#### 8.2.2. SQL Server Profiler

SQL Server Profiler は、SQL Server に付属しているユーティリティで、DBMS が実行するステートメント、実行の内部処理を経過時間などをトレースします。本来はパフォーマンスチューニングのためのツールですが、 DBMS で実行される SQL 文をリアルタイムで観察することができるために、デバッグのツールとしても有効です。

下図は、デフォルトの設定で uniPaaS から APG でテーブル参照を行ったときのトレース結果の一部です。ここで見るように、SELECT 文の発行、レコードのフェッチ、トランザクションの開始、終了などが行われたことがわかります。

Note: Sold Server Profiler - [無題 - 1 (AREYOMI)]													
눩 ファイル(F) 編集(E) 表示(\	/) 再生(R) ツール(T) ウィンドウ(W)	へ,レプ(H)				_ d	F ×						
🖹 🔁 🚔 🔉 🖉 🖉 🖉 🖉 👘 🖉 📄 🥵 🔂 📰 🚳 🖗													
EventClass	TextData	ApplicationName	NTUserNa	LoginName	CP	Reads	۱.						
SQL:BatchCompleted	SELECT csid FROM master.dbo.syschar	uniPaaS	isoda	areyom	0	0	-						
SQL:BatchStarting	select @@version	uniPaaS	isoda	areyom									
SQL:BatchCompleted	select @@version	uniPaaS	isoda	areyom	0	0							
SQL:BatchStarting	select id from MAGICsysobjects wh	uniPaaS	isoda	areyom									
SQL:BatchCompleted	select id from MAGICsysobjects wh	uniPaaS	isoda	areyom	0	3							
SQL:BatchStarting	set implicit_transactions on	uniPaaS	isoda	areyom									
SQL:BatchCompleted	set implicit_transactions on	uniPaaS	isoda	areyom	0	0							
RPC:Completed	declare @p1 int set @p1=180150003	uniPaaS	isoda	areyom	0	0							
RPC:Completed	exec sp_cursorfetch 180150003,32,1,1	uniPaaS	isoda	areyom	0	3	Ξ						
RPC:Completed	exec sp_cursorfetch 180150003,32,1,1	uniPaaS	isoda	areyom	0	2							
SQL:BatchStarting	IF @@TRANCOUNT > 0 COMMIT TRAN	uniPaaS	isoda	areyom									
SQL:BatchCompleted	IF @@TRANCOUNT > 0 COMMIT TRAN	uniPaaS	isoda	areyom	0	0							
SQL:BatchStarting	set implicit_transactions off	uniPaaS	isoda	areyom									
4		·n 0			^	-							
decire @pi int set @pi = R01 int set @pi = R0150003 decire @p3 int set @p3-2 decire @p4 int set @p4-1 decire @p5 int set @p5 in													
トレースを実行しています。				行 18、列	2	行:25							
				接続数: 1									

設定を変えることにより、トレースする情報を細かく指定することができます。詳しくは SQL Server Profiler のへ ルプなどを参照してください。



uniPaaS ではサーバカーソルを利用しているために、Profiler 上ではストアドプロシージャへの呼び 出しを行う、一見複雑な SQL 文が発行されています。例えば、次のようなものです。

declare @p1 int	
set @p1=180150003	
declare @p3 int	
set @p3=2	
declare @p4 int	
set @p4=1	
declare @p5 int	
set @p5=-1	
exec sp_cursoropen @p1 output,N'SELECT ID,LN,PRICE FROM MAGIC.dbo.MI_DTL ORDER BY	
D ASC ,LN ASC ',@p3 output,@p4 output,@p5 output	
select @p1, @p3, @p4, @p5	

この SQL 文は、SQL Server のライブラリが作成しているもので、uniPaaS の SQL Server ゲーにも 説明があります。トウェイが実際に発行している SQL 文は、最後から 2~3 行目にある以下の SELECT 文です。

SELECT ID,LN,PRICE FROM MAGIC.dbo.MI_DTL ORDER BY ID ASC ,LN ASC

Profiler を使って、uniPaaS が発行している SQL 文を確認したい場合には、このことに留意してください。



SQL Server Profiler は、無償の Express Edition には含まれていません。Developer Edition などを ご利用ください。

# 第9章 uniPaaS 製品以外のツール

本章では、uniPaaSのデバッギングに役立つ、サードパーティ製品やフリーソフトなどのツールの使い方を説明します(下表)。

種類	機能	設定
プロセス状態	各プロセスの CPU/メモリ等の利用状況の 表示	Windows タスクマネージャ、 リソースモニタ
システム イベント ログ	Windows レベルでのイベントの記録	Windows システムイベントログ
HTTP 通信	HTTP 通信データのキャプチャ	横取り丸
ネットワーク通信	ネットワーク通信キャプチャ	ネットワークモニタ
キーボードマクロ	キーボード/マウス操作の記録と再生	(各プログラム)

# 9.1. Windows タスクマネージャ

Windows のタスクマネージャは、Windows に標準に装備されているユーティリティで、現在実行中のプロセスに 関する情報およびシステムの状況をリアルタイムで表示することができます。

#### 9.1.1. STEP 1. タスクマネージャを起動する

タスクマネージャは、デスクトップのタスクバーで 右クリックメニューから「タスクマネージャ」を選択 することにより起動します。



初期画面は次のように、現在デスクトップ上に出 ているプログラムの一覧が表示されます。

📕 Windows タスク マネージャ	
ファイル(E) オブション(O) 表示(V) ウィンドウ(W) シャットダウン(U) ヘル:	プ(日)
アプリケーション プロセス パフォーマンス ネットワーク ユーザー	
タスク 状態	
②uniPaaS Studio V1 Plus     実行中	•
C¥ 実行中	2
	>
タスクの終了(E) 切り替え(S) 新しに	はスク( <u>N</u> )
	208KB

## 9.1.2. STEP 2. プロセス一覧を見る

Windows で実行されているプロセスは、「アプリケー ション」タブで一覧表示されているものばかりではあ りません。内部的にバックグラウンドで多くのプロセ スが実行されています。それを見るには、「プロセ ス」タブを押します。一覧の「イメージ名」タイトルを マウスでクリックすると、イメージ名でソートされるの で、目的とするタスクを探しやすくなります。

	Windows タスク マネ (川(E)、オポション(O)	ー <b>ジャ</b> まニ0.0	20mm上が白5.700 /	× 11 =9/11)	(	. 🗆 🗙		
77 7	かりケーション プロセス	3€.1.(¥) )(73-5	クネットダークし エークーク	-#-				
	イメージ名	PID	ユーザー名	セッション	C	火단 🔼		
(	svchostexe svchostexe svchostexe System Jele Proce Jaskmer eve uniRQBroker exe uniRQBroker exe uniRTExe VmwareService exe VMwareTray exe VMwareTray exe VMwareTray exe windogomexe	696 764 812 876 4 0 800 1132 2004 3920 <b>3884</b> 1384 392 380 664 420	NETWORK SERV SYSTEM NETWORK SERV LOCAL SERVICE SYSTEM Administrator Administrator Administrator Administrator SYSTEM Administrator Administrator Administrator Administrator SYSTEM		00 00 00 99 00 00 00 00 00 00 00 00 00 0	2		
	<					>		
□全ユーザーのプロセスを表示する(S) プロセスの終了(E)								
プロł	27:30 CPU 使序	<b></b> ]]本: 0%	コミット チャージ	: 293412KB /	490208	кв		

uniPaaS で使うプログラムの主なものは次の通りです。

uniPaaS 製品	モジュール名
開発版(uniPaaS Studio)	uniStudio.exe、uniRTE.exe
実行版 (Magic Client)、および uniPaaS サーバ(Enterprise, RichClient)	uniRTE.exe
リッチクライアント クライアント側モジュール	uniRC.exe
Magic リクエストブローカ	uniRQBroker.exe

#### 9.1.3. STEP 3. 表示項目を追加する

「プロセス」タブのデフォルトの設定では

- ・ イメージ名
- ユーザ名
- CPU 利用%
- が各プロセスごとに表示されますが、より詳細な情報を知るために、これに加えて
  - ・ プロセス ID
  - メモリ使用量
  - 仮想メモリサイズ

も表示させるようにします。

このほか、プロセスの活動状況をうかがうには、

- I/O 読み取りバイト数
- I/O 書き込みバイト数

なども役に立ちます。

メニュー「表示」→「列の選択」メニューを選択します。

マネ	ージャ		
D	表示(⊻)	シャットダウン(山)	へル
2	最新の 更新の	情報に更新( <u>R</u> ) 頃度( <u>U</u> ) ▶	ューザ
(	列の選	IR(S)	C
	LOCAL	L SERVICE	-

右図のような「列の選択」ダイアログが表示されます ので、必要な項目を選択してください。 この設定はレジストリに記憶されるので、以後、タスク マネージャを再起動しても有効です。

列の選択	
タスク マネージャの [プロセス	] ページに表示する列を選択します。
<ul> <li>イメージ名(1)</li> <li>PID (プロセス ID)(P)</li> <li>CPU 使用率(C)</li> <li>CPU 時間(E)</li> <li>メモリ使用量(M)</li> <li>メモリ使用量(M)</li> <li>メモリ使用量(M)</li> <li>ストメモリ使用量(K)</li> <li>ページフォルト(E)</li> <li>USER オブジェクト(U)</li> <li>I/O 読み取り</li> <li>ゾロ(K)</li> <li>セッション IO(S)</li> <li>マーザー名(M)</li> </ul>	<ul> <li>ページ フォルト デルタ(A)</li> <li>✓ 仮想メモリ サイズ(V)</li> <li>ページ プール(G)</li> <li>事ページ プール(Q)</li> <li>基本優先度(B)</li> <li>ハンドルの数(H)</li> <li>スレッドの数(D)</li> <li>GDI オブジェクト(J)</li> <li>IVO 書き込み</li> <li>ジバク 書き込み,</li> <li>ジバク 香き込み,</li> <li>IVO その他</li> <li>IVO その他のバイト数</li> </ul>
	OK ++>セル

## 9.1.4. システムの全体的な状況を見る

システムの全体的な状況は、「パフォーマンス」タブ で見ることができます。ここでメモリや CPU の利用 状況を確認することができます。

Windows タスク マネ	-94		
ァイル(E) オブション( <u>O</u> )	表示(⊻) シャット	ダウン(山) ヘルプ(田)	
アプリケーション プロセス 🤇	パフォーマンスネ	ットワーク ユーザー	
-CPU 使用率	- CPU 使用率の	履歴	
0 %			
	~ページ ファイル(	使用量の履歴	
287 MB		~物理 \于! (KB)	
ハンドル スレッド	7538 397	合計 利用可能	523760 263080
7022	30	システム キャッシュ	254820
 合計 制限値 最大値	294312 490208 423112	カーネル メモリ (KB) 合計 ページ 非ページ	27376 22692 4684
			1000001/17
1世ス: 30 CPU 使)	円半:∪% □	ミット チャージ: 294312KB 7 /	490208KB

## 9.1.5. プロセスを強制終了させる

プログラムがハングアップしたり無限ループに陥ったりしてしまったときに、どうしても停止させることができない 場合には、最後の手段として、タスクマネージャからプログラムを強制終了させることができます。

この方法を使うと、必要な終了処理などが一切行われずにプログラムが強制的に終了させられるので、データの破損や他のプログラムの動作への影響などが起こる可能性があります。したがって、プログラムの強制終了 はあくまで、やむを得ず行う最後の手段としてください。 プロセスを強制終了させるには、タスクマネージャ の「プロセス」タブを開き、強制しようとするプログラ ムにカーソルを合わせ、「プロセスの終了」ボタン を押します。

	Windows タスク マネー	-97				
77	イル(E) オブション( <u>O</u> ) ま	表示(⊻) う	ノャットダウン(山) ヘル	ブ( <u>H</u> )		
P	プリケーション プロセス ア	フォーマンフ	、 ネットワーク ユーサ	ř–		
		/				
	イメージ名	PID	ユーザー名	C	メモリ使	<u> </u>
	svchost.ex svchost.exe svchostexe svchostexe Systen Systen Idle Process taskiner.exe	696 764 812 876 4 0 800	NETWORK SERV SYSTEM NETWORK SERV LOCAL SERVICE SYSTEM SYSTEM Administrator	00 00 00 00 98 00	4,664 K 20,996 K 3,648 K 4,800 K 260 K 28 K 7,488 K	
0	unin Broker.exe unin QBroker.exe uniRTE.exe	1132 2004 3920	Administrator Administrator Administrator	00	6,836 K 2,204 K 26,404 K	
	uniStudioexe VMwareService exe VMwareTrayexe VMwareUserexe w3dbsmgrexe	3884 1384 392 590 664	Administrator SYSTEM Administrator Administrator Administrator	00 00 00 00	4,428 K 2,192 K 3,376 K 5,944 K 24,812 K	=
	winlogon.exe	420	SYSTEM	ŐŐ	4,144 K	-
		 表示する( <u>S</u>	Ű	C	りロセスの終了(」	
プロ・	セス: 30 CPU 使用	]率: 2%	コミット チャージ: 29	4284KE	3 / 490208KB	

# 9.2. リソースモニタ

リソースモニタはタスクマネージャと同じく、Windows に備わっている Microsoft 社のユーティリティで、タスクマ ネージャよりもより細かく、プロセス単位で CPU 利用率、メモリ利用量、ディスク IO、ネットワーク活動などをリア ルタイムで監視することのできるユーティリティです。

Windows Vista 以降の Windows (Windows Vista、Windows Server 2008、Windows 7) では、タ スクマネージャから「リソースモニタ」を起動するこ とができます。 タスクマネージャを起動し、「パフォーマンス」タブを 開き、「リソースモニタ」ボタンを押します。 コマンドラインから、「resmon」を実行しても、起動 させることができます。



リソースモニタが開きます。



#### 9.2.1.「CPU」タブ

「CPU」タブを開くと、各プロセスごとの CPU 利用率、スレッド数等が表示されます。

🔕 リソース モニ	9-										
ファイル(F) 監	俔(M) 🗸	√レプ(H)									
概要 CPU	メモリ	ディスク ネットワー	ウ								
- TX-9	PID	説明	状態	スレッド	CPU	平均 CPU	ユーザー名	管理者特権		$\mathbf{\delta}$	<u> </u>
svchost.exe	980	Windows サービ	実行中	12	25	0.54	SYSTEM	はい			1000/
🔲 java.exe	7612	Java(TM) 2 Platf	終了済み	17	2	0.77	SYSTEM	はい		CPU - Ast	100% 7
oracle.exe	2476	Oracle RDBMS Ke	実行中	30	2	0.09	SYSTEM	はい	=	~~~~	~~~~~
perfmon.exe	5508	リソースとパフォ	実行中	19	1	2.73	isoda	はい			
perl.exe	6508	perl.exe	終了済み	1	0	0.19	SYSTEM	はい		60 秒	0%
perfmon.exe	7508	リソースとパフォ	実行中	19	0	0.41	isoda	はい		サービス CP	U使用率 100% J
📄 sqiservr.exe	1964	SQL Server Wind	実行中	37	0	2.07	NETWOR	はい			
🔲 dwm.exe	3736	デスクトップ ウィ	実行中	5	0	0.26	isoda	はい			4
Smc.exe	1164	Symantec CMC S	実行中	32	0	0.23	SYSTEM	はい	=		0%
System	4	NT Kernel & Syst	実行中	127	0	0.17	SYSTEM	-		CPUID	100% -
📄 システムの	-	遅延プロシージャ	実行中	-	0	0.16	SYSTEM	-			10070
SnagIt32.exe	2580	SnagIt 8	実行中	5	0	0.12	isoda	はい			
📄 synergys.exe	4364	synergys.exe	実行中	7	0	0.10	isoda	はい		~~~ <b>*</b> *	
📄 java.exe	2948	Java(TM) 2 Platf	実行中	45	0	0.08	SYSTEM	はい			0% _
TNSLSNR.E	1660	Oracle SQL*Net	実行中	5	0	0.06	SYSTEM	はい		CPU 1	100%
csrss.exe	480	クライアント サー	実行中	20	0	0.05	SYSTEM	はい			
📄 emagent.exe	3056	Oracle Enterprise	実行中	9	0	0.04	SYSTEM	はい			
svchost.exe	940	Windows サービ	実行中	20	0	0.04	LOCAL SE	はい			0%
vmware-v	4564	VMware Workstat	実行中	8	0	0.03	isoda	はい		CPU 2	100% 7
explorer.exe	1136	エクスプローラー	実行中	22	0	0.03	isoda	はい			
vmware-au	2944	VMware Authoriz	実行中	7	0	0.03	SYSTEM	はい			
BuildServic	3708	IncrediBuild Age	実行中	13	0	0.03	SYSTEM	はい			
svchost.exe	1588	Windows サービ	実行中	18	0	0.02	LOCAL SE	はい			0% _
CSrss.exe	400	クライアント サー	<u> 実行由</u>	10	0	0.01	SYSTEM	はい		CPU 3	100% т 🔻

## 9.2.2. 「メモリ」タブ

「メモリ」タブを開くと、システム全体のメモリ使用量とその内容、および各プロセスごとのメモリ使用量の内訳を 見ることができます。

©												
ファイル(F) 些 <mark>現(H)</mark> ヘルプ(H)												
概要 CPU	メモリ	ディスク ネットワーク										
プロセス 📕 物理メモリ 61% 使用 📀 🏠											21- <b> </b> -	
🔲 イメージ	PID	ハード フ	コミット (	ワーキング	共有可能 (	プライベー		^		使用物理メモリ	100% ¬	1
oracle.exe	2476	0	493,708	484,432	55,364	429,068						
svchost.exe	980	0	220,236	218,072	9,360	208,712						
📄 java.exe	2948	0	160,980	165,836	11,172	154,664						
📄 sqlservr.exe	348	0	196,960	126,708	32,428	94,280						
📄 soffice.bin	3044	0	89,124	151,508	64,408	87,100						
📃 dwm.exe	3736	0	81,952	162,032	90,556	71,476				60 秒	0% _	1
🔲 uniRTE.exe	6808	0	49,784	71,944	28,356	43,588				コミット チャージ	100% -	1
vmware.exe	7572	0	47.276	63.352	25.056	38.296		*				
物理メモリ 📕 5037 MB を使用中						📕 3122 M	4B を使用可能					
									Ε			Ξ
	ドウェア系	約済 📕 使用	ф.	■ 変更済。	4	スタンバイ	空き	-			0%	
		503	7 MB	32 MB	,	3082 MB	40 MB			ハード フォールト/5	むんし 和 100-	
1 MB												
				利用可能	3122 M	1B						
				キャッシュ済	み 3114 M	1B						
				合計	8191 M	1B						
				インストール	済み 8192 №	1B					ا وهجروز	
								,			0 _	
												-11
									Ŧ			~

## 9.2.3. 「ディスク」タブ

「ディスク」タブを開くと、各プロセスごとのディスクIOの活動状況、およびシステムの各論理ディスク(ドライブ)ごとの活動状況が表示されます。
ג-עע 🚳	モニター													<u>α</u> Σ	×
ファイル(F)	監視(M)	ヘルプ(H)													
概要 CPU	J XE	ディスク チットラ	リーク												
🔲 イメージ			PID	読み取り (/	(イト/秒) 書き	込み (バイ	合計 (/ (イ	′ト/秒)			<u> </u>	$\odot$	- ² 7	_	] ^
System			4		11,742	174,720	18	6,462				7 5		N4D (\$4)	4
GoogleIM	EJaConver	ter.exe	4708		2,731	30,379	3	3,109				71,20	1	MB/#9 -	
oracle.exe	2		2476		15,560	11,641	2	7,201			-				
soffice.bir	ı		3044		5,274	21,844	2	7,118				A			
System			0		0	1,638		1,638				60 秒		0 _	
emagent.	exe		3056		0	1,155		1,155			-	ディスク0	(C: U:) の	. 0.01	
ディスク活	昏動	<mark>=</mark> र्न.	ィスク I/O 916	KB/秒		📕 アクティブ	な時間の最	高 0%		Ó					
イメージ	PID	ファイル		読み取り (	書き込み (	合計 (バイ.	I/O 🕼	そうしょう しんしょう しんしょう しんしょう しんしょう しんしん しんしん し	お答時間(ミリ	J	<u>^</u>	Ac	ANA.		
System	4	C:¥Users¥isoda¥AppD	ata¥Local¥	0	968,448	968,44	8	標準		4				0 _	
System	4	C:¥Users¥isoda¥AppD	ata¥Local¥	2,731	322,816	325,54	7	標準		6		ディスク1	(F:) のキュ.	0.01	
System	4	C:¥Users¥isoda¥AppD	ata¥Local¥	60,621	59,750	120,37	1	標準		16	Ε				
System	4	C:¥Users¥isoda¥AppD	ata¥Local¥	0	100,160	100,16	0	標準		14		A			
System	4	C:¥Users¥isoda¥AppD	ata¥Local¥	8,192	77,248	85,44	0	標準		4.	-			0	
記憤城											5	ディスク 3	(E:) のキュ・	1	
		******			SUBJECT OF AN ADDRESS				5 .h.						
調理ティスク		物理ティスク	アクティン	(30時間 (96)	利用可能/よ税職(	(MB) ±	∺ңляқ (МВ)	712	<i>⊘</i> ∓⊥		- 11				
F:		1		0.00	203,	485	423,564		0.00					0	
C: U:		0		0.00	/33,	418	953,866		0.00			ディスク 2	(X: D:) の	, 0.01 ך	
E: X. D.		3		0.00	78,	823	253,999		0.00						
X. U.		2		0.00	/41,	092	900,667		0.00						
												A.		0	
											-				-

## 9.2.4. 「ネットワーク」タブ

「ネットワーク」タブを開くと、各プロセスごとの、TCP/IPによる通信活動状況、利用しているポート番号、接続の確立したコネクションの情報が表示されます。

🔕 リソース モニイ	9-											x
ファイル(F) 監	視(M) /	√レプ(H)										
概要 CPU	メモリ	ディスク ネット	ワーク )	)								
ネットワーク	活動の	プロセス							•	۲	E⊐-  ▼]	14
🔲 イメージ	PID	送信 (パイ 受	信 (バイ	合計 ()(イ					•	ネットワーク	10 Kbps –	
java.exe	2948	200	96	297					E			
emagent.exe	3056	147	135	283								
svchost.exe	1208	0	243	243								
BuildServic	3708	139	25	164					-	· · · · · · · · · · · ·	441	
	7000	22		124						PANT L		
ネットワーク	活動	📕 ネ	・ットワーク I	I/O 6 Kbps			■ ネットワー	・ク使用率 0%	)	60 秒	L 0	
TCP 接続										TCP 按抗	100 -	
イメージ	PID	ローカル アドレス	<b>—</b>	リモート アド	レス	IJ€	パケット損	潜在期間 (	<u> </u>			ш
System	4	10.3.0.109	445	10.3.0.227		2195	0	140				
synergys.exe	4364	10.3.0.109	24800	10.3.0.227		2110	0	10				
uniRTE.exe	6808	10.3.0.109	52521	10.3.0.11		1521	0	1			الككككك	
BuildService.exe	3708	10.3.0.109	49168	10.3.0.49		31104	0	0			L0	
java.exe	2948	IPv4 ループバック	49175	IPv4 ループハ	(ック	49176	0	0	*		接続 100% -	
リッスン ポー	1								•			
イメージ	PID	アドレス	ポート	プロトコル	ファイ	アウ			^			
System	4	IPv6 が未指定です	80	TCP	不許可	·					أكككككك	
System	4	IPv4 が未指定です	80	TCP	不許可	·						
svchost.exe (	800	IPv6 が未指定です	135	TCP	許可、	制					0 _	
svchost.exe (	800	IPv4 が未指定です	135	TCP	許可、	制				VMware Netwo	ork A 100% -	
System	4	192.168.73.1	137	UDP	不許可	k						
System	4	192.168.67.1	137	UDP	不許可	k						
System	4	10.3.0.109	137	UDP	不許可	·					الا کا کا ک	
Svstem	4	192.168.73.1	138	UDP	不許可				 <b>v</b>			-

# 9.3. システムイベントログ

## 9.3.1. システムイベントログとは?

システムイベントログとは、Windows オペレーティングシステムが備えるログ機構で、「コンピュータの管理」ユー ティリティの一部として統合されています。ここには、Windows がさまざまなアプリケーションを実行していくうち に遭遇するイベントを記録していきます。

uniPaaSを実行しているうちに、プログラムの実行に異常があった場合にも、異常についての簡単な情報がシステムイベントログに記録されます。

**閉く(の)** エクスプローラ(<u>メ</u>)

検索(E)...

<del>ネットワーク</del>ネライブの割り当て(<u>N</u>).

ネットワーク ドライブの切断(1)...

ショートカットの作成(S) 削除(D) 名前の変更(M) プロパティ(B)

管理(G)

## 9.3.2. システムイベントログの使い方

システムイベントログは、特に設定をしなくとも、デフォ ルトで有効になっています。

システムイベントログを見るには、まず、「コンピュータ の管理」ツールを開きます。これは、デスクトップの「マ イコンピュータ」アイコンを右クリックして開くメニューか ら「管理」を選択します。



-					
📙 コンピュータの管理					
🗐 ファイル(E) 操作( <u>A</u> ) 表示	この ウルド	が(W) ヘルプ(H)			_ <del>_</del> _ <del>/</del> ×
← → 🗈 💽 🗗 🖸	3 🔮 🖬				
🗐 コンピュータの管理 (ローカー	種類	日付	時刻	ソース	分業 🛆
	③情報	2008/07/09	14:26:54	DrWatson	なし 👝
	8 <u>1</u> 5-	2008/07/09	14:26:51	Application Error	なし
	③情報	2008/07/09	14:26:14	SecurityCenter	なし
Internet Expl	③情報	2008/07/09	14:26:13	VMTools	なし
10 ビイエクノイ	③情報	2008/07/09	12:36:11	SecurityCenter	なし
	③情報	2008/07/09	12:36:10	VMTools	なし
■ ● □ーカル フーザー	③情報	2008/07/09	12:32:50	MsiInstaller	なし
■ 刷パフォーマンス □グ	③情報	2008/07/09	12:32:50	MsiInstaller	なし
二月 デバイスマネージ	③情報	2008/07/09	12:32:50	LoadPerf	なし
□ 🔤 記憶域	③情報	2008/07/09	12:32:49	LoadPerf	なし
🚡 🤮 リムーバブル記憶	③情報	2008/07/09	12:32:49	LoadPerf	なし 💌
<	<				>

uniPaaSでエラーが起こった場合には、「アプリケー ション」に、IISでエラーが起こった場合には「システ ム」のイベントに記録されることが多いようです。 IISのエラーは、「ソース」として「WAM」あるいは 「W3SVC」という名前となります。

昌 コンピュータの管理					
📃 ファイル(E) 操作( <u>A</u> ) 表示	: : : : : : : : : : : : : : : : : : :	♡(₩) ヘルプ(出)			- 8 ×
← → 🛍 🖬 🗗 🛙	2 🛛 🕄 🖬				
🚚 コンピュータの管理 (ローカ🔺	種類	日付	時刻	ソース	分数 🔼
💷 🌇 システム ツール	③情報	2010/09/06	16:42:14	Service Control Manag	なし
	③情報	2010/09/06	16:42:14	Service Control Manag	なし
III Internet Evel	③情報	2010/09/06	16:42:14	Service Control Manag	なし
	③情報	2010/09/06	16:42:14	Service Control Manag	なし
E 2777	▲警告	2010/09/06	16:42:14	W3SVC	なし
● 共有フォルダ	③情報	2010/09/06	16:41:45	eventlog	なし
■ 3 ローカル フーザー	③情報	2010/09/06	16:41:45	eventlog	なし
🛓 📓 パフォーマンス ログ	🔅 情報	2010/09/06	16:40:55	eventlog	なし
ニュニュ デバイス マネージ・	🔅 情報	2010/09/06	16:38:31	Service Control Manag	なし
🖻 🚵 記憶域	🔅 情報	2010/09/06	16:38:31	Service Control Manag	なし
🔲 🚡 🤮 リムーバブル記憶 ⊻	③情報	2010/09/06	16:38:31	Service Control Manag	なし 💌
< >	<				>

# 9.4. その他のシステムモニタツール

#### 9.4.1. Process Explorer

Process Explorer は、Sysinternals (http://technet.microsoft.com/ja-jp/sysinternals/default.aspx) からダウン ロードできるユーティリティの一つです。

Process Explorer は、Windows のタスク マネージャより細かな情報を見ることが できます。各プロセスごとに、プロセスツ リーや、イメージ・メモリ・ネットワーク・ ディスク等の活動状況の詳細をリアル タイムで見ることができます。

ile <u>O</u> ptions <u>V</u> iew <u>P</u> rocess F	ind <u>U</u> sers <u>H</u> elp				
🖬 🕼 🔳 🖩 🐼 😭	X 🗛 🕀	:			
Process	PID CPU	Private Bytes	Working Set	Description	Company Name
🜍 explorer.exe	1 4 4 0	15,392 K	6,720 K	Windows Explorer	Microsoft Corporation
💮 VMware Tray.exe	1828	1,340 K	3,376 K	VMware Tray	VMware, Inc.
- VMwareUser.exe	1844	2,156 K	5,928 K	VMwareUser	VMware, Inc.
📝 otfinion .exe	1856	1,436 K	3,732 K	CTF Loader	Microsoft Corporation
💦 msmsgs.exe	1864	1,940 K	2,392 K	Windows Messenger	Microsoft Corporation
ooo w3db smare∞e	1884	8,736 K	13,804 K	Database Service Manager	Pervasive Software Inc.
Desktops.exe	720	2,944 K	6,140 K	Sysinternals Desktops	Sysinternals – www.sysinter
🥥 explorer.exe	564	12,360 K	18,088 K	Windows Explorer	Microsoft Corporation
👸 explorer.exe	1156	12,356 K	18,024 K	Windows Explorer	Microsoft Corporation
👸 explorer.exe	536	12,360 K	18,016 K	Windows Explorer	Microsoft Corporation
SystemExplorer.exe	800	11,452 K	12,524 K	System Explorer	Mister Group
🚰 procexp.exe	264	13,468 K	16,848 K	Sysinternals Process Explorer	Sysinternals – www.sysinter
🖃 🦰 uniStudio.exe	164	56,196 K	4,092 K	uniPaaS Studio	Magic Software Enterprises
uniRTE.exe	612	20,760 K	26,328 K	uniPaaS runtime	Magic Software Enterprises
🔤 🏞 uniRQBroker.exe	724	784 K	2,196 K	Requests Broker	Magic Software Enterprises
uniRQBroker.exe	1040	4,204 K	6,744 K	Requests Broker	Magic Software Enterprises

#### 例1: uniPaaS 実行エンジンのメモリ消費の推移を見る

実行中の uniPaaS 実行エンジン (uniRTE.exe) のメモリ消費量は、システム安定稼働に影響を与えます。タスク マネージャなどでも各プロセスのメモリ使用量の瞬間値、およびシステム全体の使用量の推移は見ることがで きますが、各プロセスごとのメモリ使用量の推移は見ることができません。Process Explorer を使えば、各プロ セスごとにメモリ消費量の推移を見ることができます。

例えば、uniRTE.exe のメモリ消費量を見るには、次のようにします。

Process Explorer を開いて、uniRTE.exe を探します。

Process Explorer – Sysin	ternals: www.s	sysinternals.com	[RDWINXP¥Ad	mini 💶 🗖	l
File Options View Process	Find Users H 7 🛪 🏘 🥸				
Process	PID CPU	Private Bytes	Working Set	Description	
🗉 🦳 System Idle Process	0 98.46	OK	28 K		
😑 🧊 explorer.exe	1272	13,704 K	24,004 K N	Vindows Explorer	
🕞 VMware Trayexe	1 472	1,340 K	3,376 K N	/Mware Tray	
VMwareUser.exe	1480	2,160 K	5,964 K N	/MwareUser	
📝 otfmon .exe	1488	1,444 K	3,752 K C	OTF Loader	
💦 msmsgs exe	1500	1,936 K	2,460 K N	Vindows Messenger	
≝ w3dbsmgrexe	1536	82,028 K	24,904 K E	atabase Service M	Is
🖃 🥁 Marne 5.exe	600	6,316 K	3,580 K 🛔	b的File5	
 procexp.exe	1832 1.54	16,764 K	9,648 K S	ysinternals Proces	5:
🖃 🖉 uniStudio exe	3096	56,960 K	2,448 K u	iniPaaS Studio	
	3052	126,556 K	134,864 K u	iniPaaS runtime	ł
	2276	784 K	2,200 K F	Requests Broker	
niRQBroker.exe	2292	4,192 K	6,788 K F	Requests Broker	
Beloo					
	<			>	
OPU Usage: 1.54% Commit Charg	e: 32.34% Proces	sses: 28 Physical Usa	age: 68.24%		

ダブルクリックしてプロセスのプロパティ ダイアログを開き、Performance Graph タブを開きます。

右の図は、メモリテーブルに大量のレ コードを作成していったときの推移です。 上から、CPU 利用率、メモリ使用量、 ディスクへの入出カバイト数の推移を 表したグラフです。



## 例2: mglock.dat を掴んでいるプロセスを探す

uniPaaSの実行中に、自分以外は誰も使っていないはずなのにファイルやレコードロックが発生する、ということがあります。原因はいろいろ考えられますが、uniPaaS独自のロック機能を実現している mglock.dat ファイルを、他のプロセスがつかんだままになっている、という可能性があります。このような場合に、どのプロセスがmglock.datをオープンしているかを調べられると便利です。

Process Explorer を使うと、特定のファイル (ファイル名の一部だけでも良い)をオープンしているプロセスを探し出すことができます。

- 3. Process Explorer を開きます。
- メニュー「Find → Find Handler or DLL」を選びます。Process Explorer Search ダイアログが開き ます。

🤰 Process Explorer – Sysie	ternale	<del>: www.sysinterna</del> ls	.com [RDWINX	P¥Adm 💶 🗖	X
<u>F</u> ile <u>O</u> ptions <u>V</u> iew <u>P</u> rocess	Find <u>U</u> s	ers <u>H</u> elp			
: 🖬 😰 💻 🖹 🚟 😽 👔	<u>F</u> ind Ha	andle or DLL Otrl+F			
Process	CPU	Private Bytes	Working Set De	escription	^
🛨 🔤 System Idle Process	96.92	ΟK	28 K		
😑 👮 explorenexe		11,640 K	18,612 K. Win	dows Explorer	
🕞 VMware Tray.exe		1,340 K	3,376 K. VM	ware Tray	
MwareUser.exe		2,048 K	5,096 K. VM	wareUser	
📝 otfmon exe		1,436 K	3,708 K OTF	- Loader	
🚔 w3dbsmgr.exe		82,008 K	24,844 K Dat	abase Service Mana <b>g</b> e	er 🔤
🖃 🧑 uniStudio.exe		55,748 K	2,300 K unif	PaaS Studio	
🔄 🕐 uniRTE.e.xe		22,604 K	3,984 K unif	PaaS runtime	
ouniRC exe		9,060 K	9,844 K unif	PaaS Rich Client	
🖃 🚰 uniRQBroker.exe		784 K	2,204 K Red	quests Broker	
🐂 uniRQBroker.exe		4,192 K	6,764 K Red	quests Broker	
🖃 📸 Mame5.exe		5,800 K	3,364 K まめ	File5	
<b>Dy</b> proce xp.e xe	1.54	14,580 K	18,136 K Sys	internals Process Exp	oloi
~					$\sim$
	<			) (	>
CPU Usage: 3.08% Commit Charge	: 24.99%	Processes: 30 Physic	al Usage: 40.68%		

 「Handle or DLL substring」欄に、 探したいファイル名 (今の場合 mglock.dat)を指定し、「Search」ボ タンを押します → 結果が表示され ます。



 ダブルクリックします → ダイアログ が閉じ、そのファイルをオープンし ているプロセスに位置づけられま す。



System - Sysin	ternals: «	www.sysinternals	s.com [RDWINXP¥Adm 📳 🗖	×
<u>F</u> ile <u>O</u> ptions <u>V</u> iew <u>P</u> rocess	F <u>i</u> nd H <u>a</u> no	ile <u>U</u> sers <u>H</u> elp		
: 🖬 😰 📰 🖬 🖬	7 🔨 🏘	1 🔮 🕴		1
Process	CPU	Private Bytes	Working Set Description	^
📝 otfmon exe		1,436 K	3,708 K CTF Loader	
🚔 w3dbsmgr.exe		82,008 K	24,844 K Database Service Manage	r 🗐
🖃 🧑 uniStudio exe		55,732 K	3,852 K uniPaaS Studio	-
( OuniRTE.exe		22,604 K	3,996 K uniPaaS runtime	
uniRCass		9,060 K	9,844 K uniPaaS Rich Olient	~
🖃 🔂 uniRQBroker.exe	<			
Type + bleme				
i ype – Name				
File C:¥Program Files¥i	JniPaaS¥Stu usiPaaS¥Stu	idio VI Plus¥Projects¥ idio VI Plus¥Projects¥	MyApp1 ¥Source MyApp1 ¥melor lor	
File C:¥WINDOWS¥Win	nn ⊨aas≢stu SxS¥x86 Mio	crosoft.VC80.CRT 1 fc8	www.appr≆melog.og b3b9a1e18e3b80.50727.3053 x−ww.b8	
File C:¥Documents and	l Settings¥A	Il Users¥Application Da	ata¥Pervasive Software¥PSQL¥logs¥pvs	
File C.¥WINDOWS¥WIN	SxS¥x86jvik	prosoft.vG80.CRT_I fc8	b3b3al e1 8e3b_8 0 597 27 3053 x-ww.b8	-
File C:¥Program Files¥	uniPaaS¥Stu	idio VI Plus¥Projects¥	MyApp1 ¥m¢lock.dat	
	A CONTRACTOR AND A	denin interation 36 density of the	DotoWMineson@WMM_IP9_1Wimin91.u.dia	
File C:¥Doouments and	10010085 <del>0</del> /1	anning a group phogue	n batatmiorosotteniilor oʻji timipor a dic	
File C:¥Doouments and Key HKLM	Difference fil	(We down NTXOurrest)	Versiee #Drivers20	~
File C:#Decourternts and Key HKLM Key HKLMSOFTWARI	E¥Microsoft¥	(Windows NT¥Current)	Version #Drivers32	~

#### 例3: ロードされている DLL を確認する。

システム上に同一名の DLL がある場合、設定などの間違いにより、名前は同じだが、意図していたものと異なる DLL がロードされて、動作が意図したとおりにならないことがあります。

たとえば、異なったバージョンの uniPaaS をインストールしていると、同一名のゲートウェイ DLL が異なったバージョンのインストールディレクトリに存在します。異なるバージョンのゲートウェイがロードされると、動作が微妙に異なったり異常終了したりしま。

また、「外部コール U=UDP」コマンドで呼び出す、ユーザ定義プロシージャの DLL として、同名だが異なったものがロードされることもあります。いずれもこのような状況は、なかなか原因をみつけにくい問題になります。 このような問題は、ProcessExplorerを使うと、特定のプロセスでロードしている DLL の一覧を見て確認することができます。 ProcessExplorer を起動し、確認したい プロセスを見つけます。 右図では、Studio のプロセス uniStudio.exe を選択しています。 実行時の動作の不正に関する場合に は、実行モジュール uniRTE.exe を選択 します。

メニューから、 View → Show Lower Pane、 を選択した後、 View → Lower Pane View → DLLs を選択します。右図のように、プロセス がロードしている DLL の一覧が表示さ れます。

DLLのパスが表示されていない場合に は、タイトル行 (「Name」など)で右クリッ クしてポップアップメニューから「Select Columns」を選択し、ダイアログで 「Path」にチェックを入れます。 (この設定は、次回以降にも記憶されま す)



File Options Vie	ew Process Find D	LL U	sers I	Help			
	🖪 🥵 😭 メ 🛤	1					A.,
Process		PID	CPU	Description	Compan y	Private Byt	Virtual Size
My proce:	хрб4.ехе	4284	0.76	Sysinternals Proc.	Sysinterna	41,396 K	159,444 K
🧱 Hide maru	exe	6964		秀丸エディタ	有限会社	5,416 K	278,708 K
😑 🦰 uniStudio	.exe	5324		uniPasS Studio	Magic Soft	113,680 K	349,208 K
👝 🧑 uniRTI	Elexe	5620		uniPasS runtime	Magic Soft	24,768 K	177,436 K
_ <u></u>	non-tracting	4004		Desire de Dostres	Marca	1.1.00 K	EE LOO M
Name	Description	0	ompan y l	Name V	ersion		
Ib.IOC	Oracle Call Interface	01	acle Cor	poration 11	02.0000.0001		
zlibwapi.dll WPDEF32.DLL	zlib data compression libr	ary		10	0000, 1000, 20		
uniStudio.exe PDFCreatorPilot3.dll	uniPaaS Studio	Ma	așic Soft	ware Enterpris… 1 ມ	0000.0000.00		
MS VOR71 .dll	MicrosoftR C Runtime Li	ibra Mi	crosoft (	Oorporation 7.1	0.3052.0004		
matxerces−o_2_6 dll MGVC dll	Shared Library for Xerce	s Ap	ache So	ftware Founda 2.0	0000.0000.000		
MgS vn taxEditor dll	uniPaaS Rich Client	Ma	ric Soft	vare Enterpris 1.0	8 0001 0387		



Name または Path でソートして、間違っ た DLL がロードされていないかを確認 してください。 右図の例では、MGORACLE.DLL が異

なったインストールディレクトリ D:¥MAGIC¥150SP1 からロードされてい ることがわかります。

これは、MAGIC.INI の [MAGIC_GATEWAYS] の設定で誤った DLL が指定されているときに起こります ので、MAGIC.INI を再確認することで対 応する、ということになります。





Process Explorer、および下記の Process Monitor, VmMap を含め、Sysinternals から提供される ユーティリティをすべてまとめたものとして、「Sysinternals Suite」が無償でダウンロードできます。 (http://technet.microsoft.com/ja-jp/sysinternals/bb842062.aspx)

## 9.4.2. Process Monitor

Process Monitor も Sysinternals から提 供されるユーティリティで、各プロセスに ついて、ファイルアクセス、レジストリ操 作、ネットワーク活動などを、オペレー ション単位で監視することができます。

💐 Process Monitor -	Sysinternals: www.sysinternals.com		
<u>F</u> ile <u>E</u> dit E <u>v</u> ent Fi <u>l</u> ter	Tools Options Help		
😂 🖬   🍳 📴 🖾	🗢 🗛 🐵   🗛 🖡   🎎 🔜 🛝 🗤 🌆		
Tim Process Name	PID Operation Path	Result	Detail 🗠
12:09: ⊘uniStudio.exe	3116 💁 QueryDirectory C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Filter: Prg_2
12:09: 🕝uniStudio.exe	3116 🔜 CloseFile C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	
12:09: 🧑uniStudio.exe	3116 SOreateFile C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Desired Acc
12:09: 🧑uniStudio.exe	3116 🛃 QueryDirectory: C#Program Files¥uniPaaS¥Studio:V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Filter: Prg_2
12:09: 🥝uniStudio.exe	3116 KoloseFile C. * Program Files * uniPaaS * Studio V1 Plus * Projects * MyApp1 * Source	SUCCESS	
12:09: 🕝uniStudio.exe	3116 🕏 CreateFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Desired Acc
12:09: 🎆uniStudio.exe	3116 🦣 ReadFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Offset: 0, Le
12:09: 🕜 uniStudio.exe	3116 ReadFile C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Offset: 0, Le
12:09: 🧑uniStudio.exe	3116 ReadFile C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	END OF FILI	EOffset: 5,34
12:09: 🧑uniStudio.exe	3116 🔜 CloseFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	
12:09: 🧑uniStudio.exe	3116 🕏 CreateFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Desired Acc
12:09: 🧑uniStudio.exe	3116 💁 QueryDirectory: C. ¥Program Files¥uniPaaS¥Studio: V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Filter: Prg_2
12:09: 🧑uniStudio.exe	3116 SCloseFile C. ¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	
12:09: 🧑uniStudio.exe	3116 🔜 ReadFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥uniStudio.exe	SUCCESS	Offset: 2,69!
12:09: ÖuniStudio.exe	3116 ReadFile C#Program Files¥uniPaaS¥Studio V1Plus¥uniStudio.exe	SUCCESS	Offset: 2,61
12:09: ÖuniStudio.exe	3116 💁 ReadFile C: ¥ Program Files¥uniPaaS¥Studio V1 Plus¥uniStudio.exe	SUCCESS	Offset: 2,66:
12:09: ÖuniStudio.exe	3116 CreateFile C: Program Files ¥uniPaaS ¥Studio V1 Plus ¥Projects ¥MyApp1 ¥Source	SUCCESS	Desired Acc
12:09: 🧑uniStudio.exe	3116 SQueryDirectory C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	Filter: Prg_2
12:09: 🧑uniStudio.exe	3116 CloseFile C#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1¥Source	SUCCESS	~
2000 <b>–</b> 10. II		~	- 0// · 0 00/
Showing 13,100 of 77,670 eve	nts (16%) Backed by page file		

#### 利用例1: MAGIC.INI の確認

uniPaaS 起動時に、標準の uniPaaS ディレクトリ上にある MAGIC.INI を使うのではなく、別の箇所にある MAGIC.INI を指定して起動することがあります。また、コマンドラインに @(ファイル名) の形式で、追加設定を指 定することもできます。

このときに、ちゃんと指定しているはずであるにもかかわらず、思ったように設定がされていない(論理名が定義されていない、など)という場合があり、この場合には「どこの MAGIC.INI を参照しているのか?」あるいは「@ (ファイル名)で指定したファイルがちゃんと読み込まれているのか?」ということを確認したくなります。 このような場合に、Process Monitor のファイルアクセス活動を監視すると、チェックすることができます。 例えば、ショートカットを作成し、「リンク先」として、

"C:\Program Files\UniPaaS\Studio V1Plus\UniRTE.exe" /ini=MYMAGIC.INI @ENV\MGADD.INI

と指定した uniRTE.exe を起動したときに、設定ファイル MYMAGIC.INI および ENV¥MGADD.INI が正しく読み込 まれているかを確認するには、次のようにします。

Process Monitor を起動し、メニュー 「Filter → Filter」を選択します。



「Image Path」「is」「C:¥Program Files¥uniPaaS¥Studio V1Plus¥ uniRTE.exe」 then「Include」として、 「Add」ボタンを押します。 この設定は、記憶されます。

Process Moni	tor Filter				
Display entries match	ing these conditions				
Image Path	🗸 is 🗸	C:¥Program Files¥ur	niPaaS¥Studio V1Plus	¥uniRTE.exe 🔽 the	en Include 💙
Reset				Add	<u>R</u> emove
Column	Relation	Value	Action		^
🥝 Image Path	is	C:¥Program F	Include		
😵 Process Na	is	Procmon.exe	Exclude		
😵 Process Na	is	System	Exclude		
😢 Operation	begins with	IRP_MJ_	Exclude		
🔇 Operation	begins with	FASTIO_	Exclude		
😢 Path	ends with	pagefile.sys	Exclude		~
			<u></u> K	<u>C</u> ancel	Apply

ツールバーで、「Show File System Activity」(右図のアイコン)のみを有効 にしておきます。

作成した uniRTE.exe のショートカットを 起動します。uniRTE.exe が起動して、 Process Monitor にファイル IO 活動の ログが記録されます。

最初に、MYMAGIC.INI が正しく読まれ ているかを調べます。 メニュー「Edit → Find」で、 MYMAGIC.INI を指定します。

Result が SUCCESS となっているので、 正しく読み取れていることがわかります。 その後のログを見ても、CreateFile (ファ イルオープン)、ReadFile (ファイル読み 込み) など SUCCESS (最後は END OF FILE) になっていることを確認します。

ENV¥MGADD.INI についても、同様に調 べます。

<u>File Edit Ev</u> ent Fi <u>l</u> ter <u>T</u> ools <u>O</u> ptions <u>H</u> elp		
😅 🖬   🔍 📴 🖾   ኞ 🔺 🌚   🛤 🦐   🎎 🗟 🛝 🗤 🌆		
Tim Process N PID Operation Path	Result	Detail 🗠
16:37: 🟉 uniRTE.exe 🔰 1884 🛃 QueryNameInfC:¥Program Files¥uniPaaS¥Studio V1Plus¥uniRTE.exe	SUCCESS	Name: ¥Pro
16:37: 🧑uniRTE.exe – 1884 🛃QueryNameInfC:¥Program Files¥uniPaaS¥Studio V1Plus¥uniRTE.exe –	SUCCESS	Name: ¥Pro
16:37: 🥝uniRTE.exe 1884 🔂 CreateFile C:¥WINDOWS¥Prefetch¥UNIRTE.EXE-18215182.pf	SUCCESS	Desired Acc
16:37: 🥝uniRTE.exe 🛛 1884 🛃 QueryStandar C:¥WINDOWS¥Prefetch¥UNIRTE.EXE-18215182.pf 👘	SUCCESS	AllocationS
16:37: OuniRTE.exe 1884 💁 ReadFile C:¥WINDOWS¥Prefetch¥UNIRTE.EXE-1B2151B2.pf	SUCCESS	Offset: 0, L
16:37: 🥝uniRTE.exe 1884 🛃 CloseFile C:¥WINDOWS¥Prefetch¥UNIRTE.EXE-18215182.pf	SUCCESS	
16:37: ⊘uniRTE.exe 1884 🛃 CreateFile C:	SUCCESS	Desired Acc
16:37: MuniRTE eve 1884 🔍 Quervinformat G:	RI IFFER ∩	_VolumeΩre:≚
<		>
Showing 3,858 of 259,042 events (1.4%) Backed by page file		

検索		? 🗙
検索する文字列(N): MYMAGICINI		次を検索( <u>F</u> )
□ 単語単位で探す(W)	検索する方向	キャンセル
□大文字と小文字を区別する(0)	○上へ(凹) ○下へ(D)	

🗌 🎪 民 👠 🧠 🗖

now File System Activit

🚔 Process Monitor – Sysinternals: www.sysinternals.com			×
<u>Eile Edit Ev</u> ent Fi <u>l</u> ter <u>T</u> ools <u>O</u> ptions <u>H</u> elp			
🖙 🖬   🛠 📴 🙄   💝 🔺 🛞   🏘 🦻   🎎 🔜 🗛 🦙 🌆			
Tim Process N PID Operation Path	Result	Detail	^
16:37: OuniRTE.exe 1884 🔂 CreateFile C ¥ Program Files¥uniPaaS¥Studio V1Plus	SUCCESS	Desired A	
16:37: 🖉 uniRTE,exe – 1884 🔜 QueryDirectory C:¥Program Files¥uniPasS¥Studio V1Plus¥MYMAGIC.NI	SUCCESS	Filter: MY	_
16:37: 🖉uniRTE.exe 1884 🔒 CloseFile 🔨 C¥Program Files¥uniPaaS¥Studio V1Plus	SUCCESS		
16:37: 🧿uniRTE.exe 1884 💁 CreateFile C:¥Program Files¥uniPaaS¥Studio V1Plus	SUCCESS	Desired Ar	
16:37: 🖉uniRTE.exe 🛛 1884 🛃 QueryDirectory C:¥Program Files¥uniPaaS¥Studio V1Plus¥MYMAGIC.NI	SUCCESS	Filter: MY	
16:37: 🖉uniRTE.exe 1884 🛃 CloseFile C¥Program Files¥uniPaaS¥Studio V1Plus	SUCCESS		
16:37: 🥏uniRTE.exe 1884 💁 CreateFile C¥Program Files¥uniPaaS¥Studio V1Plus¥MYMAGIC.ini	SUCCESS	Desired A	-
16:37' 🧖 uniRTE eye 🛛 1884 🔍 ReadFile 👘 Ci¥Program Files¥uniPaaS¥Studio V1Plus¥MVMAGIC ini	SUCCESS	Offset: 0	~
		>	
Showing 3,860 of 286,780 events (1.3%) Backed by page file			

🚔 Process Monitor – Sysinternals: www.sysinternals.com		
<u>File Edit Event Filter Tools Options H</u> elp		
🕼 🖬   🛠 🕅 🕼   🗢 🛆 🌚   🛤 🦻   🎎 🗟 🔍 🛪 🔳		
Tim Process N PID Operation Path	Result	Detail
16:37: 🗇uniRTE.exe 1884 🔂 CloseFile 🦯 C.¥Program Files¥uniPaaS¥Studio V1Plus¥MGresrc.dii	SUCCESS	
16:37: 💭uniRTE.exe 1884 💭OreateFile C:¥Program Files¥uniPaaS¥Studio V1Plus¥ENV¥MGADD.NI	SUCCESS	Desirec
16:37: 🧿uniRTE.exe 1884 💁 ReadFile 📔 C:¥Program Files¥uniPaaS¥Studio V1Plus¥ENV¥MGADD.MI	SUCCESS	Offset:
16:37: 🥏uniRTE.exe 1884 💁 ReadFile 📔 C:¥Program Files¥uniPaaS¥Studio V1Plus¥ENV¥MGADD.NI	SUCCESS	Offset:
16:37: 🧿uniRTE.exe 1884 💁 ReadFile 📔 C:¥Program Files¥uniPaaS¥Studio V1Plus¥ENV¥MGADD.NI	END OF FILF	Offset:
16:37: 🧿uniRTE.exe 🛛 1884 🛃 CloseFile 🔪 C:¥Program Files¥uniPaaS¥Studio V1Plus¥ENV¥MGADD.NI	SUCCESS /	
16:37: 🥭uniRTE.exe 🛛 1884 💁 CreateFile 🗡 O:¥Program Files¥uniPaaS¥Studio V1Plus	SUCCESS	Desired
16:37' - 👰 iniRTE eye - 1884 🖳 QueryDirectory C:¥Program Files¥umiPaaS¥Studio V1Plus¥MVMAGIC INI	SLICCESS	Filter: N
		>
Showing 3,860 of 286,780 events (1.3%) Backed by page file		

## 利用例2:書き込んだはずのファイルが作成されていない

バッチタスクでテキストファイルを出力したはずなのだが、ファイルが作成されていない、というような場合に、原 因を追求するために、ProcessMonitorを使うことができます。

出力 – NA

⊘入出力ファイル: 7 -

16:54: 16:54

検索

検索する文字列(<u>N</u>):

単語単位で探す(W)

□大文字と小文字を区別する(C)

Showing 845 of 26,377 events (3.2%)

NA.TXT

例えば、「%MYTMPDIR%NA.TXT」という ファイルにデータを出力するバッチタス クを実行するとします。

前記の例と同様にして、Process Monitor で、uniRTE.exe の ファイル IO 活動をモニタさせます。 この状態で、uniPaaSのバッチタスクを

実行します。 Process Monitor に、IO 活動のログが 記録されます。

Process Monitor - Sysinternals: www.sysinternals.com File Edit Event Filter Tools Options Held i 🖉 🚰 2 🍺 💟 🗟 🗛 😔 🛯 🎎 😹 😹 😹 約 🥇 Tim Process N... PID Operation Path Result Detail UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe UniRTE.exe SUCCESS SUCCESS 3116 3116 C:¥Program Files¥uniPaaS¥Studio V1Plus¥Suppor CreateFile Desirec Filter∶ι QueryDirecto 16:54 C#Program Files¥uniPaaS¥Studio V1Plus¥Support¥usr_std 16:54: 3116 🕄 Close File C:¥Program Files¥uniPaaS¥Studio V1Plus¥Support SUCCES: 3116 CreateFile 3116 QueryDirecto 3116 CloseFile 3116 CreateFile 3116 ReadFile 16:54: C#Program Files¥uniPaaS¥Studio V1Plus¥Support SUCCESS Desired 16:54: C#Program Files¥uniPaaS¥Studio V1Plus¥Support¥usr std SUCCESS. Filter: ι 16:54: C#Program Files¥uniPaaS¥Studio V1Plus¥Support SUCCESS

Backed by page file

検索する方向 ●上へ(山) ○下へ(山)

C#Program Files¥uniPaaS¥Studio V1Plus¥Support¥usr_std. C#Program Files¥uniPaaS¥Studio V1Plus¥Support¥usr_std.

2

次を検索(F)

キャンセル

SUCCESS

SUCCESS

Desire

Offer

「NA.TXT」をキーワードをして検索をし ます。

この例では、Result が PATH NOT FOUND となっていました。 パス名を見 てみると、C:¥tmpp¥NA.TXT となってい ました。

C:¥tmp ディレクトリに書き込むつもり だったのであれば、ディレクトリ名が間 違っていた、というこになります。結局、 ファイルが作成されなかった原因は、論 理名 MYTMPDIR の定義の間違い、と いうことになります。

Process Monitor – Sysinternals: www.	sysinternals.com	(	
<u>File Edit Event Filter Tools Options H</u> el	p		
🖙 🖬   🍳 🏁 🖾   🗢 🔺 🛞   d	🍇 📕 📔 🎎 🄜 🤐 🌆		
Tim Process N PID Operation	Path	Result	Detail 🔷
16:54: OuniRTE.exe 3116 NotifyChange 16:54: OuniRTE.exe 3116 NotifyChange 16:54: OuniRTE.exe 3116 NunlockFileSin	C¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1 C¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1 <mark>©#Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp</mark>	SUCCESS SUCCESS SUCCESS	Filter: F Filter: F Offset:
16:54: OuniRTE.exe 3116 CreateFile 16:54: OuniRTE.exe 3116 CreateFile 16:54: OuniRTE.exe 3116 NotifyChange	C¥tmpp¥NA.TXT C¥tmpp¥NA.TXT O <del>¥Program Filos¥uniPasS¥Studio V1Plus¥Projects¥MyApp1</del>	PATH NOT. PATH NOT. SUCCESS	. Desirec . Desirec Filter: F
16:54: ÖuniRTE.exe 3116 NotifyChange 16:54: OuniRTE.exe 3116 UnlockFileSin	C¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp1 C¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyApp	SUCCESS	Filter: F
Showing 845 of 69,500 events (1.2%)	Backed by page file		

フィルタとしては、次のような設定が便利と思います。

UniPaaS 関連のプロセスを監視するには、

[Company] [contains] [Magic Software Enterprises]

- 特定のモジュール (例えば uniRTE.exe)を監視するには、
  - ▷ 「Image Path」「is」「(uniRTE.exe のフルパス名)」、あるいは
  - Process Name J [is ] [uniRTE.exe]
- 実行中の特定のインスタンスを監視するには、
  - ▷ 「PID」「is」「(プロセス ID)」



ProcessMonitor は Windows の API レベルのログをとるので、簡単な操作でも大量のログが作成さ れ、実行速度にも影響を与えます。従って、ProcessMonitor を使った調査は、実運用に影響を与 えないテスト環境で、十分に問題が絞り込まれてから行うようにしてください。



<u>F</u> ile <u>E</u> dit E <u>v</u> ent Fi <u>l</u> ter <u>T</u> ools <u>O</u> ptions <u>H</u> elp		
🖙 🖬   🔍 🎯 🖾   💎 🔺 🎯   👫 🦐   🎎 🗟 🗛 🧊 🛄		
Tim Process N PID Operation Path	Result	Detail 🐴
16:54: 🍘uniRTE.exe – 3116 🛃 NotifyChange C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyA	pp1 SUCCESS	Filter: F
16:54: 🥝uniRTE.exe 🛛 3116 💁 NotifyChange C.¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyA	pp1 SUCCESS	Filter: F
16:54: 🖉uniRTE.exe 🔄 3116 🛃 Unlock FileSin 🖓 🕸 Program Files ¥uniPaaS¥Studio V1Plus ¥Projects ¥MyA	ppSUCCESS	Offset:
16:54: 🖉uniRTE.exe 3116 🛃OreateFile 🚺 O:¥tmpp¥NA.TXT	PATH NOT,	Desirec
16:54: 🌍uniRTE.exe 3116 💁.CreateFile 🔍 C.¥tmpp¥NA.TXT	PATH NOT.	Desirec
16:54: 🥏uniRTE.exe – 3116 <mark>B</mark> NotifyChange 🕽 <del>XProgram FilesYuniPaaSYStudie V1PlusYProjectsYMyA</del>	pp1 SUCCESS	Filter: F
16:54: 🌍uniRTE.exe – 3116 🛃 NotifyChange C:¥Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MyA	.pp1	Filter: F
1654: 🖉 uniRTE eye – 3116 🖳 Uninck Eile Sin – Gi¥ Program Files¥uniPaaS¥Studio V1Plus¥Projects¥MvA	nn SHOOESS	Offset: 🎽
		>
Showing 845 of 69,500 events (1.2%) Backed by page file		

## 9.4.3. VmMap

vmmapは、プロセスが使っているメモリ 内容をかなり細かく見ることができます。 メモリリークが疑われる場合に詳細に 調査することができます。

VMMap - Sy	sinternals: www	.sysinter	nals.com						
<u>F</u> ile <u>E</u> dit <u>R</u> efre	esh Options <u>H</u> elp								
Process: PID:	uniRTE.exe 612								
Committed:							_		82,224 K
Private:									20,760 K
Working Set:									26,348 K
Туре	Size	Co	mmitted	Private	Total V	VS I	Private WS	Shareable 1	WS Shar
Total	121,212 K		82,224 K	20,760 K	26,348	3 K	15,988 K	10,36	i0 K
Image	62,948 K		62,948 K	4,692 K	18,833	2 K	9,120 K	9,71	2 K
Mapped File	712 K		712 K		68	вк		6	i8 K
Shareable	7,120 K		2,496 K		576	6 K		57	'6 K
Неар									
Managed Heap	00.400 K		144 1	144 K	10.		104 8		
Stack	20,480 K		144 K	144 K	104	4 K	104 K		4 V
Private Data	22,204 K		6,170 K	6,170 K	0,380		0,370 K		4 K
Linknown	7.260 K		7.260 K	7.960 K	000	DK	300 K		
Free	1,983,624 K		7,000 K	7,000 K					
	<								>
Address +	Туре	Size	Commi	Private	Total WS	Privat	Share	Sha Blog	ke Protec 🔨
00010000	Densete Dete	10120	× V		10(01 100		ondro	010 0100	1 0
00010000	Private Data	4 N.	4 N.	4 N	4 N	4 N 4 K			1 Read/W
m 00020000	Thread Stack	2048 K	60 K	60 K	56 K	56 K			3 Read All
00230000	Shareable	12 K	12 K	00 K	8 K	00 K	8 K	8 K	1 Read
00240000	Shareable	8 K	ι 8 κ		8 K		8 K	• •	1 Read
± 00250000	Private Data	1,024 K	304 K	304 K	184 K	184 K			2 Read/W
⊕ 00350000	Private Data	64 K	24 K	24 K	24 K	24 K			2 Read/W
⊕ 00360000	Shareable	64 K	12 K		8 K		8 K	8 K	2 Read/W
00370000	Mapped File	88 K	88 K		20 K		20 K	20 K	1 Read
00390000	Mapped File	260 K	260 K		12 K		12 K	12 K	1 Kead 🗸
003E0000	<								>
Snapshot: 11:15:0	0								

#### 9.4.4. SystemExplorer

SystemExplorer も、システム診断のた めのユーティリティで、タスクマネー ジャよりも詳細な情報を表示すること ができます。非常に多くの機能があり ます。

http://www.systemexplorer.net/から ダウンロードすることができます。



#### 9.4.5. Process Walker

やました工房(http://www001.upp.so-net.ne.jp/yamashita/index.htm ) からも、システムやプロセスの状況監 視ツールが公開されていますが、ProcessWalker Express が多くの情報を収集することができ、便利と思います。 このユーティリティでも、各プロセスに関して非常に多くの情報を表示させることができます。

💐 ProcessWalker Express			
: ファイル(E) 編集(E) 表示(V) ツール(D ウ	ィンドウ(型) ヘルプ(日)		
🗣 🗣 😼 🧬 🔍 🖆 🚺 🖡	× 🖬 🖬 🧟		
コンソールエクスプローラ × 🗇 IPアドレ	3		
くしてく していた。 この、 この、 この、 この、 この、 この、 この、 この、	/F91929270-5		
	Desktons.exe : 720	▲ 「リイントリの」「首報	
	AM フロセス		
	名前 PID 🔺 ユーザー	ワーキングセット	🏽 プライベートン 🔼
	🥃 explorer.exe 1440 RDWINXP¥	Administrator 16,015,360	16,592
	VMwareService.exe 1508 NT AUTHO	DRITY¥SYSTEM 2,506,752	913
	YUM exe I /64 RDWINXP¥	Administrator IU,U72,U64	4,300
	Windre frayexe     1626 RDWINAF#     1626 RDWINAF#     1844 RDWINXP#	Administrator 5,041,344	2 203
	Ctfmon.exe 1856 RDWINXP¥	Administrator 4,009,984	1,478
	🔏 msmsgsexe 1864 RDWINXP¥	Administrator 3,047,424	1,986 🔽
			>
	ハンドル		×
	種類 ハンドル 🔺 オブジェクトア	7ド 名前	<u>~</u>
	🕼 KeyedEvent 0x4 0xE10096E0	) ¥KernelObjects¥CritSecOutOfMemory	Event
, and the second s	Directory 0x8 0xE159BD50	0 ¥KnownDIIs	
	File     OxC 0x81F96F90     OxC 0x81F96F90     OxC 0x81F96F90	) ¥Device¥HarddiskVolume1¥Program	Files¥uniPaaS
	Semaphore UX10 UX8228A7D8	5 VUR.J	✓
	×		>
	🗞 モジュール 📲 スレッド 🔬 ハンドル		
< · · · · · · · · · · · · · · · · · · ·			
36個のプロセス.312個のハンドル			

# 9.5. パケットキャプチャプログラム

#### 9.5.1. 横取り丸

横取り丸というのは、HTTPプロトコルに基づく通信の要求と応答の内容を記録しておくフリーウェアのユーティ リティです。Web対応アプリケーションや、uniPaaSリッチクライアントシステムの動作に問題がある場合、原因 調査に役立ちます。

横取り丸は、ベクター(http://www.vector.co.jp/)などのダウンロードサイトや、開発者のサイト

(http://hidemaru.xaxon.co.jp/index.html) から無料でダウンロードすることができます。

横取り丸は、プロキシサーバとして動作し、パケットのデータを保存します。一覧には、リクエスト時刻、応答時 刻、メソッド、リクエスト・応答サイズなどが表示されます。

<b>鄂 InetSpy - ログ取り中</b>							
ファイル(E) 設定(S) ヘルブ(H)							
URL	りケエスト時刻	転送時刻	応答時刻	終了時刻	メソット	リクエストサイス゛	応答サイズ
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?RICHCLIENT=INITIAL&QU%26Ae%7c0sirisc	13:47:31.39	+0.00	+3.37	+3.37	GET	349	435
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?RICHCLIENT=Y&UTF8TRANS=Y&CTX=167	13:47:34.85	+0.00	+0.03	+0.04	GET	207	257
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	13:47:34.93	+0.00	+2.23	+2.26	POST	311	3722
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.28	+0.00	+0.03	+0.03	GET	185	16934
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.39	+0.00	+0.01	+0.01	GET	189	2050
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.40	+0.01	+0.03	+0.03	GET	189	2050
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.43	+0.00	+0.01	+0.01	GET	179	7935
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.48	+0.00	+0.01	+0.01	GET	187	1316
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.51	+0.00	+0.01	+0.01	GET	208	1582
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.64	+0.00	+0.01	+0.03	GET	182	42051
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.70	+0.00	+0.03	+0.06	GET	203	6523
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX=16723528963584&CACHE=agent_3684	13:47:37.90	+0.00	+0.01	+0.01	GET	234	5158
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	13:48:16.46	+0.00	+0.06	+0.07	POST	1126	499
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	13:48:17:50	+0.01	+0.03	+0.04	POST	1150	499
http://rdwinxp/uni18Scripts/Mgrqispi018.dll	13:48:18.60	+0.00	+0.03	+0.04	POST	1150	499

各リクエストをダブルクリックすると、その内容が表示されます。 また、キャプチャしたデータはファイルに保存し、後からその ファイルを開いて調査することもできます。

リクエスト情報	応答情報 時刻など	[]			
応答情報( <u>F</u>	): サイズ =	499bytes			
HTTP/1.12 Server: Micr Date: Wed. 3 X-Powered Content-Ty uniRTEExec uniBrokerW Content-Ler (?xml id="Mwrtro <language (xuserRim)<global (xml)<global (xml)<global (xml)<global (xml)<global (xml)</global </global </global </global </global </language 	00 OK soft-IIS/51 10 Oct 2010 04:48:16 By: ASP NET betext/html utionTime: 47,0 utionTime: 47,0 utinTime: 0 gth: 299 pn="10"?> DATA"> DATA"> DATA"> itTime: 0 gth: 299 pn="10"?> DATA"> DATA"> itTime: 0 gth: 299 pn="10"?> DATA"> itTime: 0 gth: 299 pn="10"?> itTime: 0 gth: 299 pth:	GMT '/> /globalParamsCl	nanges>	<dataview task<="" td=""><td>id=".</td></dataview>	id=".
<					>
中身を閲	覧ソフトで見る( <u>V</u> )	中身をファー	イルレに保存( <u>S</u> )		
		OK	4.0	Setzili I	∧ 11.⊐?

#### 例1: リッチクライアント起動時のエラー

リッチクライアント関係の設定が間違って いると、リッチクライアントプログラムを起動 する際にエラーが出ます。 例えば、「リモートサーバがエラーを返しま した」が出る場合を見てみます。

横取り丸で通信を監視すると、右図のよう なパケットでした。





中身を見てみると、応答の結果が、「404 Object Not Found」でした。 これは、指定された URL を、Web サーバが 認識できなかったことを意味します。



そこで URL を再度良く見てみると、仮想 ディレクトリが uni18Script になっていまし た。本当は uni18Scripts (最後の「s」があ る)なので、これが誤りであることがわかり ます。

この仮想ディレクトリは、MAGIC.INI ファイ ルの InternetDispatcherPath から取られる ものなので、MAGIC.INI を確認し、修正しま す。



ファイル(E) 編集(E) 書式(Q) 表示(V) ヘルブ(H) WebDocumentPath = C:¥Program Files¥uniPaaS¥Studio V1Plus¥RIAModules InterpathicroatherPath (mil2Sociat Alexainsi018 d)	
WebDocumentPath = C:¥Pro <del>gram Files¥uni</del> PaaS¥Studio V1Plus¥RIAModules	
WebDocumentAlias = /unit8 <del>RIAModules</del>	<b>^</b>
	¥

## 例2: ブラウザコントロールやイメージが表示されない

リッチクライアントプログラムの起動は OK だが、ブラウザコントロールに PDF ファイ ルや HTML ファイルを表示する際にエラー になる場合や、イメージファイル(背景、イ メージコントロール、イメージボタン) が表 示されないなどの問題がある場合を見て みます。

このようなエラーが出る場合には、しばらく (1~2分)だんまりになって、その後にエ ラーとなる、という場合もしばしばあります。

横取り丸で見てみると、次の ような応答が見られます。 「応答サイズ」に表示がなく、 ダブルクリックして「応答情 報」を見ても空白でした。

park:	
i	へ Webサイト側でページを表示できませれ
	可能性のある原因: • Web サイトがメンテナンス中である。 • Web サイトに、プログラム上の問題がある。
	対処方法:
<	

🚮 InetSpy - 口灯取り中								×
ファイル(E) 設定(S) ヘルブ(H)								
URL	りケスト時刻	転送時刻	応答時刻	終了時刻	メソット	リクエストサイズ	応答サイズ	^
http://rdwinxp/uni18Scripts/Mgrqispi018.dll	15:19:17.96	+0.00	+0.01	+0.04	POST	1119	1032	_
http://rdwinxp/uni18Scripts/Mgrqispi018.dll?CTX	15:19:18.01	+0.01	+0.03	+0.06	GET	203	4429	
http://myserver01/tmp/%90%BF%8B%81%8F%91_2	15:19:18.18			+7.09	GET	351		
http://rdwinxp/uni18Scripts/Mgrqispi018.dll	15:19:27.60	+0.01	+0.03	+0.04	POST	1381	503	_
Htp://rdwinxp/uni18Scripts/Mgrqispi018.dll	15:19:37.76	+0.00	+0.01	+0.04	POST	1096	1032	
<pre>{http://myserver01/tmp/bill_20101021_153030.PDF</pre>	15:19:37.82			+2.26	GET	337		-

このような状況は、URLに指定されたサーバ名が認識できなかったか、指定の Web サーバが起動していなかった場合に起こります。今の例では、URL は http://myserver01/… でしたが、この myserver01 の Web サーバが認識されなかったということです。

ネットワークの設定の問題になるので、(1) myserver01 が起動しているか、(2) ping myserver01 で返答が返っ

てくるか、(3) Web サーバが起動しているか?などを確認してください。サーバの移行や、システムの移行などで、設定と実際のホスト名とが異なった可能性も考えられます。

#### 例3: 起動・動作が遅い

リッチクライアントプログラムが動くは動くのだが、起動・動作が異常に遅いので、原因を突き止めたいというこ とがあります。リッチクライアントプログラムのパフォーマンスに関しては、多くの要因がありますが、その一つと して、クライアントとサーバ間の通信量や回数が多い、という可能性があります。これについて確認するときにも、 横取り丸が使えます。

ここの例では、2項目だけの簡単なリッチク ライアントプログラムですが、コントロール 検証でわざと多数のサーバアクセスが発 生するようなロジックを入れてみました。

<b>0</b> 9:	🧼 身天夕 5 ー RichClient NA									
デー	タビュ	- ロジック	フォーム							
М	1 🖂	C=3)10-6	¥=検証	a)/kt N						
С	2	フェロック	₩=While	3 {LoopCounter() <= 10						
S	3	그네	S=#ブタスク	1 batch1						
С	4	アクション	E=式	1 CHeight('N',0)						
S	5	그네	S=#ブタスク	2 batch2						
С	6	アクション	E=式	2 CWidth('N',0)						
C	7	フェロック	N=End	}						

起動して、項目間を移動すると、もたつき 感があります。

RichClient NA	×
N: 1	
A: aaabbc	

横取り丸を見てみると、カーソル項目 N から A に移動するたびに、多くの通信が起こっている状況が見て取れます。 カーソルが項目 A から項目 N に移動する際にはこのような通信はありませんでした。

🚟 InetSpy – ログ取り中								×
ファイル(E) 設定(S) ヘルブ(H)								
URL	1	リクエスト時刻	転送時刻	応答時刻	終了時刻	メソッド	リクエスト	~
http://rdwinxp/uni18Scripts/Mgrgispi018.dll		15:33:53.64	+0.00	+0.03	+0.03	POST	1035	_
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	1	15:33:53.67	+0.00	+0.03	+0.03	POST	1035	
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	1	15:33:53.70	+0.00	+0.03	+0.03	POST	1035	
http://rdwinxp/uni18Scripts/Mergispi018.dll	1	15:33:53.73	+0.00	+0.03	+0.03	POST	1035	
http://rdwinxp/uni18Scripts/Mergispi018.dll	1	15:33:53.76	+0.00	+0.03	+0.03	POST	1035	
http://rdwinxp/uni18Scripts/Mgrqispi018.dll	1	15:33:53.79	+0.00	+0.03	+0.03	POST	1035	_
http://rdwinxp/uni18Scripts/Mgrgispi018.dll	1	15:33:53.82	+0.00	+0.03	+0.03	POST	1035	
http://rdwinxp/uni18Scripts/Mergispi018.dll	J	15:33:53.85	+0.00	+0.03	+0.03	POST	1036	~
1. 7/1 · 7 300 · At · 300 III	/	1000000	.0.00	.0.00	-0.00	DOCT	1000	

このように、横取り丸で通信の状況を観察することにより、「遅い」と感じられる場所で、裏でどのような処理が 起こっているのかを、推測することができます。確認すべきは、次のようなところでしょう。

- 通信の回数:回数が多いと動作が遅くなります。この場合不要なサーバアクセスが発生しないよう、プロ グラムロジックを見直す必要性があります。
- 「応答時刻」「終了時刻」: 1回のリクエストの応答時間に時間がかかると、全体の時間も遅くなります。応 答時間のかかっているリクエストに対応するコマンドを特定し、ロジックを見なおしてください。
- ●「リクエストサイズ」および「応答サイズ」: リクエストのデータ量が多いと、特に細い回線を使っている場合には、処理が遅くなります。データサイズを少なくするようにプログラム上の工夫をする必要があります。 (例: チャンクサイズの調整など)

また、「例2: ブラウザコントロールやイメージが表示されない」であったように、PDF ファイルやイメージファイル の URL が間違っていると、サーバからのエラーの検出に時間がかかるので、リッチクライアントプログラムの動 作が非常に遅くなってしまう場合があります。例えば、指定した URL のサーバ名が間違っていた場合、TCP/IP レベルでの名前解決のための1~2分かかり、その間、リッチクライアントプログラムはハングアップしたような 状態になります。

このような場合でも、「例2: ブラウザコントロールやイメージが表示されない」で説明したように、横取り丸で通信状況を観察すれば、問題の起こる URLを突き止めることができますので、URLを修正することで、応答速度の問題を解決できます。

## 例4: キーボード割付を取得できません



このことから、「キーボードマップテーブルなどの URL に何か問題がある」ということが推測されます。 横取り丸からは、これ以上のことを読み取ることはできませんが、キーボードマップテーブルは RIA キャッシュ に作られるものなので、RIA キャッシュに何か問題があるのではないかと思われます。 今回の例では、MAGIC.INI の RIACacheFilesPath が間違っていた(存在しない場所をさしていた)のが問題の 原因で、正しいディレクトリを指定することにより解決しました。

## 9.5.2. ネットワークモニタ

ネットワークモニタとは、ネットワーク上を流れるパケットをキャプチャするソフトウェアです。

横取り丸との大きな違いは、横取り丸は HTTP プロトコル上を流れる通信パケットに特化したユーティリティで あったのに対し、ネットワークモニタは PC 上の特定のネットワークカードを監視して、そこを通過する HTTP 以 外のプロトコルのデータも含むすべてのデータをキャプチャできるところにあります。

キャプチャするデータ量を減らすために、フィルタを定義して特定の条件が成立するパケットのみをキャプチャ するということもできます。また、キャプチャしたデータに対して、表示のフィルタをかけて、表示内容を絞り込む こともできます。

ネットワークモニタの代表的なものとしては、次のようなものがあります。

Microsoft Network Monitor は、Microsoft の Web サイト から無償でダウンロードでき るネットワークモニタです。



· Expression., Clear

192.168

.PV.-...).!..E. ....,RI CHCLIENT -WGTX-I S8198675 091576&S ESSION=6 &0ATA=% 3 ccml+id 3d\$22MGD ATA\$2283 eX0AT=% 3d\$22MGD ATA\$22MGD ATA\$2283 eX0AT=% 3d\$22MGD ATA\$22MGD ATA\$22MGD ATA

Eile Edit View Go Capture Analyze Statistics Telephony Iools Help

 Image: Constraint of the state of

28.23

eq 1

Thermet Protocol, Src: 192.168.73. A Transmission Control Protocol, Src Source port: instl_boots (1067) Destination port: http (80) [Strem index: 1] Sequence number: 5352 (relati) (Next sequence number: 6123 (n Acknowledgement number: 6358 Header length: 20 bytes # Flags: 0x18 (PSH, ACH) Andow seize: 0x660

Window size: 63683 @ Checksum: Ox1067 [validation @ [SEQ/ACK analysis] TCP segment data (871 bytes) [Reassembled TCP Segments (1035 Hypertext Transfer Protocol

File: "C#DOCLIME"1#ADMINI"1#LOCALS"1#+

WireShark は、もと Ethereal と呼ばれてい たネットワークキャプチャツールで、オープ ンソースソフトウェアです。機能としては Microsoft Network Monitor と同様で、ネッ トワーク インターフェース カードを特定し て、そこを通るデータをすべてキャプチャし ます。

Web サイト http://sourceforge.net/ projects/wireshark/ からダウンロードする ことができます。

Packetyzer は、Ethereal (現 WireShark)の 流れをくむネットワークキャプチャツールで、 操作法も似ています。WireShark に比べて、 パケットの分析機能が多くあります。

現在はアップデートが中止され、商用のソ フトウェアに吸収されたようですが、Web サイト http://sourceforge.net/ projects/packetyzer/からダウンロードす ることができます。



いずれも機能や操作法は似ているので、いずれか一つを選んで使えばよいでしょう。

# 9.6. キーボードマクロ

キーボードマクロというのは、PCユーザのキー操作やマウス操作をキャプチャ・記録して、後で同じ操作を自動 で再生することができるユーティリティです。

ー般には、決まった操作手順を自動化するために使われるものですが、テスト段階において、同じ操作を繰り返し行ないストレスをかけるようなテスト(リグレッションテスト)に応用することができます。

キー入力のシミュレーションを行うものなので、オンライン、リッチクライアントいずれのプログラムにも利用することができます。

## 9.6.1. Reckey

最初に記録を行ないます。 アプリケーションを起動し、 たい画面を表示します。

Reckey はキーボード入力のみをキャプチャ・再生することができるユーティリティです。マウス操作は利用でき ませんので、カーソルの移動には TAB キーあるいはショートカットを使い、メニューも Alt+X のショートカット キーを用いることになります。

キャプチャした内容はテキストファイルに格納されるので、適当に手で編集して、ループ構造を持つような独自のスクリプトを作成することも可能です。

機能的には比較的単純ですが、それだけに覚えるのも簡単で、単純なリグレッションテストに活用することができると思います。

RecKey は、Web ページ http://www.hi-ho.ne.jp/kyagi/ からダウンロードすることができます。

#### 利用例: アプリケーションの定形入力の繰り返し

RecKeyの利用は、(1)記録、(2)スクリプトの修正、(3)再生(実行)という3段階で行ないます。

	<b>ED_HOST</b> ファイル(E) 編録	R(D) サンブルアプリケーション	和知觉 外外型						X
ーストし		월 월 <b>  1 8 8 9 9</b> 1							
	受注番号	<u>}//)*</u> I:	在庫 💌		取引日	10/11/02	納入予定日 10/11/05		
	顧客CD				出荷先00	0			
	住那				住王所	-			
	電話番号		ファックス		電話番号		ファックス		
	顧客発注番号 担当者CD		出荷経由		撞要 決算条件CD	30 末稼めの翌	末払い		
	# 商品CD	商品名		販売数量 受注単位	課税区分CD 1	税処理CD 単価	消費税金額 金額	約期日	
	0				1 \$8067612.73				
									V
							全額		_
	<u></u>	録	受注検索(S) 印刷(	P) 削除(D) 1	@正(M) 照会	2(0) @#	k(C) 確定(C) [	<b>取消(0)</b> 終了	()()
							21		挿入

RECKEYを起動し、「記録」ボタンを 押します。



キーボードを使って、一連の処理を 行い、RecKeyに記録させます。 操作では、文字データの入力の他、 ズームからの一覧選択やファンク ションキー(F3、F4)、Enter キーなど も使うことができます。 終わったら、RecKey の記録を停止 し、適当な名前でファイルに保存し ておきます。

	IOCT												
77/1/(F)	105 T 編集	(E) サンプル	リアブリケーション	/ わりょン(0)	ウィントウへの								×
3 6	<b>II.</b> 1		相相争	₩ ☎ 🖬									
党注重号	;		) <mark>977"</mark> [	1:在庫 💌			取引日	10/11/02	纳入予定	EE 10/11/05	1		
讀客CD		0130	アールコ	Eス産業 4°aウ			出荷先CD	0 7-JJI	ス産業				
住所		476-XIOOX 愛知県 東海市南鮮	] :田町八の東城	3888			住所	476-XXXX 愛知県 東海市南些田	細いの実际に	38338			
電話番号	-	X48-285-67	54	ファックス	Х52-6Х3-К2Х	7	電話番号	X48-285-675	4	ファックス	X52-6X3-X2X7		
顧客発言 担当者の	主番号 D	123	鮎原 穂波	出荷経由	abc		摘要 決算条件CD	123 30 末締めの	D翌末払い				
# 商品	HCD HCD		商品名			販売数量 受注単位	課税区分CD	税処理CD 単	đ	消費税金額 🗟	19 I	内期日	
1 010	003		レタス			2 📵	1課税売上分	- 1	188	19	395 1	0/11/05	
2 010			レタス			3 🔞	1課税売上分		188	28	592 1	0/11/05	
学注 単号 500000026 で登録されました。													
	童師	录								金春	8 98	7	
				受注検索(§)	EUBI(6)	前膝( <u>0</u> ) 的	(F) 第	(金(Q)	登録( <u>C</u> )	確定( <u>C</u> )	取消(())	終了(	
											照会		

記録の段階で重要なことは、次のことです。

- キーボードのみで操作すること。RecKeyはマウス動作を記録できません。従って、ボタンなどはTabで パークできるようになっているか、あるいはショートカットキーでクリックすることができるようになっている 必要があります。
- 最初と最後とを合わせること。例えば、データ入力プログラムであれば、まっさらの初期状態から始まり、 一連の入力と確定とを行い、次の入力のためのまっさらの状態で終わる、というようにします。これが合っていないと、繰り返しを正しく行えなくなります。

記録が終わったら、スクリプトの編 集を行います。スクリプトの編集は、 「編集」メニューを選ぶと、ノートパッ ドが開きます。

記録直後のスクリプトは、右図のよ うなものです。 スクリプトの意味については、直観 的にも理解しやすいですが、正確に は RecKey のマニュアルを参照して ください。

100回の繰り返しを行わせるために <REPEAT 100> ··· <REND> で囲み ます。また、キー入力に適当な間隔 を置くため、<Interval 40> を入れて います。3 秒のウェイトを置きたいと ころに<Wait 300>を入れました。 編集が終わったら、ノートパッドを閉 じます。これで変更が RecKey に反 映されます。 ここで、スクリプトをまたファイルに

保存してください。

📷 ed1 - RecKey	🛛
🇅 😅 🖬 🗑 🔄 🕨 🔹	11
wn 3]{Enter]{Tab]2{Tat <mark>wic</mark> scape]{ 2KR	lend> 888
記録データを編集します	233자イト

国のよ	//*** 修正したら上書き保存してください。<< ed1-0 >>
は、直観 、正確に 参照して	<pre>{//*** C01] 00%//GIMECCC/2CC {Active ~uniPaaS RunTime~,~ED_HOST~&gt; {Tab 3}{F5}{Down 2}{Enter}{Tab 2}123{Tab}abc{Tab}qwe{Tab}{F5}{Down 2} {Enter}{Tab 2}{F5}{Down 2}{Enter}{Tab}2{Down}{F5}{Down 3}{Enter} {Tab}2{Tab}{Escape}{2}</pre>
るために 当 なた い な た い い に か に み 隔 て と し い い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い に い い に い い に い に い に い に い に い い い い い い い い い い い い い	<pre>//*** 修正したら上書き保存してください。&lt;&lt; ed1 &gt;&gt; //*** この行の次から修正してください。 <active "unipaas="" runtime","ed_host"=""> <interval 40=""> <wait 300=""> <repeat 10=""> {Tab 3}{F5}{Down 2}{Enter}{Tab 2}123{Tab}abc{Tab}qwe{Tab}{F5}{Down 2} [Enter]{Tab 2}{F5}{Down 2}{Enter}{Tab}2{Down}{F5}{Down 3}{Enter} {Tab}2{Tab}{Escape}{2} <rend></rend></repeat></wait></interval></active></pre>

以上で、スクリプトができあがりです。 あとは、「再生」ボタンを押せば、ス クリプトに従って、自動的にキー入 力が行われるようになります。



RecKey 再生時の注意事項として、次のようなことがあります。

- 再生中はマウスやキーボード等の操作を一切行ってはいけません。RecKey はスクリプトに従って、馬車馬のようにキー入力をどんどんシミュレートして行くだけであり、フォーカスがどのウィンドウにあるかとか、uniPaaSの状態がどうなっているかとかは一切わかりません。このため、マウス操作によりフォーカスのあるウィンドウが変わったりすると、意図していたようなストレステストを行うことができなくなってしまいます。
- 複雑な操作だと、スクリプトが思ったように再生されないことがあります。デバッグ機能はありませんので、 再生しながら動作を確認する試行錯誤が必要になります。タイミング的に厳しい場合には、Interval で キー入力間隔に余裕を持たせるとか、Wait で待ちを入れるなどの調整が必要になります。



#### 9.6.2. UWSC

UWSC はシェアウェア (Free バージョンもあり) のキーボードマクロユーティリティで、キー入力とマウス入力とを キャプチャ・再生することができます。http://www.uwsc.info/からダウンロードすることができます。

BASIC に似たスクリプト言語を持ち、多くの組み込み関数や制御構造があり、かなり機能の高いもので、単純な繰り返しだけでなく、条件分岐やデータを毎回変えることなどもできます。

RecKeyと比べると、RecKeyは単純な機能しかありませんが、理解しやすく手軽に使うことができます。UWSC は機能が高いので応用範囲が広いですが、理解して使いこなすのにやや壁が高いように思われます。

# 9.7. その他

## 9.7.1. Dependency Walker

Dependency Walker は、EXE や DLL モジュールの依存関係をチェックするユーティリティです。Windows の実行 可能モジュール (EXE)は、通常、多くの共有 DLL モジュールを呼び出して実行を行うので、もし利用している DLL モジュールがシステムにインストールされていなければ、起動時にエラーとなって、実行することができま せん。

uniPaaSの場合には、次のような時にこのようなエラーが起こります。

- uniPaaSモジュール (uniStudio.exe や uniRTE.exe など)が必要とする DLL が見つからないとき (誤って 削除してしまったとか、必要なサブシステムがインストールされていなかった、など)
- ゲートウェイをロードしようとしたときに、必要な DBMS クライアントモジュールがインストールされてい なかったとき。

Dependency Walker は、依存関係のある EXE/DLL ファイルをシステム内で確認し、欠損している場合にはエラーを表示します。

http://www.dependencywalker.com/ からダウンロードすることができます。

#### 例: ゲートウェイのロードに失敗する場合

MAGIC.INI ファイルで指定されているゲートウェイのロードに失敗すると、下記のようなエラーが uniPaaS 起動時に表示されます。これは Oracle のゲートウェイ (mgoracle.dll) がロードできなかった例です。

.NET-Br	oadcastEventWindow.2.0.0.0.33c0d9d.0: uniStudio.exe - コンボーネントが見つかりません	×
8	OCIdII が見つからなかったため、このアプリケーションを開始できませんでした。 アプリケーションをインストールし直すとこ 題は解決される場合があります。	の問
	<u> </u>	
🗖 エラー		I
8	Failed to load driver : C:¥PROGRAM FILES¥UNIPAAS¥STUDIO V1PLUS¥GATEWAYS¥MGORACLE.DLL	
	ОК	1

このような場合に、ゲートウェイ MGOracle.dll がロードされなかった原因を調べるために、Dependency Walker を使います。

- 1. ゲートウェイ (エラーメッセージにフルパスが記述されています。通常、uniPaaS のインストールディレクトリの下の Gateways サブディレクトリにあるはずです)が存在するかどうかを確認します。もしなければ、uniPaaS のインストーラを起動して、追加インストールします。
- 2. あるのにエラーが出ている場合には、Dependency Walker を起動します。
- 3. Dependency Walker で、ゲートウェイの MGOracle.dll を開きます。→ 次の図のような画面が表示され ます。

<mark>∎</mark> ¢ D	📭 Dependency Walker - [MGOracle.dll]									
∎¢ E	ile <u>E</u> dit <u>V</u> iew <u>O</u> ptio	ns <u>P</u> rofile <u>W</u> in	dow <u>F</u>	<u>t</u> elp					. a ×	
2	🖬 🏓 😫 🖬	E   🛺 😭   📇	<b>Q</b>	1 IV   <b>E</b>	I II   <b>№?</b>					
	MGORACLE.DLL	^	PI	Ordinal ^	Hint	Function		Entry Point		
	OCIDLL     MGXERCES-C_2.6DLL     MGXERCES-C_2.6DLL									
	EF INSERS2 DU	~	E	Ordinal ^	Hint	Function		Entry Point	~	
<		>	C#1	1(0x0001) 2(0∨0002)	0(0×0000) 1(0∨0001)	?dll_mem fm_info_e	_snapshot@@YAHH@ et	2Z 0x000044C0 0y00004370	~	
<u>^</u>	Module	File Time Stamp	i Lin	k Time Stamp	File Size	Attr.	Link Checksum	Real Checksum	CP 🔨	
?	MGXERCES-C_2_6.DLL	Error opening file	∋.指定さ	られたファイルが見	つかりません。	(2).				
2	OCI.DLL	Error opening file	き.指定さ	「れたファイルが見	つかりません。	(2).				
33	DWMAPI.DLL	Error opening file	9.指定さ	5れたファイルが見	つかりません。	(2).				
<u> X</u>	MPR.DLL	2008/04/14 11:2	5 200	08/04/14 11:25	59,904	4   A	0×0001D74A	0×0001D74A	×86	
<u> X</u>	SHLWAPI.DLL	2008/04/14 11:2	5 200	18/04/14 11:25	473,600	A	0x000759A7	0x000759A7	×86	
	ADVAPI32.DLL	2008/04/14 11:2	5 200	J8/U4/14 11:24	674,304	1 A	UXUUU AF1 B3	UXUUU AF1 B3	×86	
		2008/04/14 11:2	5 200	)8/U4/14 11:24	285,184		UXUUU4784C	UXUUU4784C	X8E	
	KERNEL32.DLL	2008/04/14 11:2	5   200	J8/U4/14 11:25	1,235,968	SIM	10X0012EF82	UXUUT2EF82	X8C 💌	
<u> </u>		11							2	
Error	Error: At least one required implicit or forwarded dependency was not found.									
Warn	ing: At least one mod	lúle has an unr	esolve	d import due	to a missin	g export	t function in a de	elay-load depende	ent n 🗡	
<u>&lt;</u>									>	
For H	elp, press F1								11	

- 4. これを見ると、「Error opening file:…」として、3つの DLL が見つからなかったことがわかります。
  - MGXERCES-C_2_6.DLL というのは、uniPaaS インストールディレクトリにあり、
     uniStudio.exe/uniRTE.exe も利用するものです。このファイルは実行時には uniStudio/uniRTE で
     ロードされているので、Gateway ディレクトリで見えなくても大丈夫です。
  - OCI.DLL というのは、Oracle のクライアントモジュールです。これが欠損しているということは、 Oracle のクライアントモジュールがインストールされていないか、インストールされていても PATH が通っていないために見つけることができなかったことを意味します。Oracle のクライアントモ ジュールを再インストールすることで解決できるかもしれません。
  - DWMAPI.DLL というものも欠損していますが、これはシステム内部で使われているものであり、また、「delay-load」という特性を持っているものなので、見つからなくても実際上は問題になりません。

# <u>第10章 プログラミングの小技</u>

uniPaaS アプリケーション実行時の問題調査やデバッグは、今まで説明してきたようなツールを使うことにより 効率化することもできますが、uniPaaS アプリケーションにちょっとした工夫をすることにより更に効率化すること もできます。

本章では、そのような uniPaaS プログラムでの小技をいくつか紹介します。



## 10.1. Logging 関数

第2章「uniPaaS 実行エンジンの出力ログ」で、アクティビティモニタによる問題追跡の方法について説明しました。アクティビティモニタに出力する内容は、「ロギングダイアログ」(2.5「ロギングダイアログでフィルタを設定するには?」参照)で設定しますが、ここで設定するとアプリケーション実行の最初から最後まで、この設定に従ってログが出力され、不要な部分のログが大量に出てきます。そうなると本当に見たいところを大量のログの中から見つけるのが難しい、実行速度が遅くなり調査に支障を来す、などが問題になることもあるでしょう。

Magic uniPaaS では、Logging 関数を使って、記録するログの内容を、プログラムから動的に変更することがで きるようになりました。これにより、ログ取りの不要な部分の実行中はログをオフにしておき、本当に必要な部分 に入ったときに有効にする、というようなログレベルのコントロールが可能になります。 この関数を実行すると、即時に変更内容が有効になります。MAGIC.INI やロギング ダイアログでの設定には影 響を与えません。

**構文:** Logging (開始/停止, フィルタ)

パラメータ:

● 開始/停止 … フィルタの開始/停止を指定する論理値

- ➢ True…フィルタを開始します。
- ➢ False … フィルタを停止します。
- フィルタ(文字)…フィルタオプションを表す文字列。以下のオプションが指定できます。

Task	Levels	DataView
Recompute	Flow	Events
LogBrowser	Gateway	TransCache
BackgroundMsg	BeginEndMsg	LogSynch
ALL	RESET	

ExecutionLogFileName=(ファイル名)

Btrieve= (N,D,S,C のどれか)

DB2400= (N,D,S,C のどれか)

Oracle= (N,D,S,C のどれか)

AS400= (N,D,S,C のどれか)

ODBC= (N,D,S,C のどれか)

MicrosoftSQLServer = (N,D,S,C o E h m)

Memory= (N,D,S,C のどれか)

- 戻り値: 論理値 … 処理が成功した場合「True」が返ります。
- 例: Logging ('FALSE' LOG,' Oracle')
- 注意事項:
- [フィルタ]の指定が正しくない場合、「False」が返ります。
  - 「ExecutionLogFileName」で外部ログファイル名が設定され、ファイルが作成できなかった場合、「False」が返ります。メッセージは mgerror.log に書き込まれます。(ログファイル名は、MAGIC.INI ファイルの GeneralErrorLog パラメータで変更できます。)
  - 外部ログが設定されず、デバッグモードも設定されていない場合、「False」が返ります。
  - [開始/停止]パラメータが「True」で DBMS パラメータが「N」と評価された場合 (例:'True','ODBC=N')、ゲートウェイオプションは無視され、「False」が返ります。

- 開発エンジンがデバッグモードでない場合、(指定されていれば)外部ログファイルのみ に反映されます。
- この関数は、MAGIC.INIを更新しません。設定された値は、現在のコンテキストでのみ 有効です。開発エンジンに戻ると、値は MAGIC.INI に設定されているデフォルト値に戻 ります。
- [フィルタ]パラメータの「ALL」が停止(False)とともに使用された場合、全ての値は False に設定されます。
- [フィルタ]パラメータの「ALL」が開始(True)とともに使用された場合、全ての非ゲート ウェイ値は True に設定されます。
- 「RESET」フィルタは、INIの値を全てリセットします。この場合、第一パラメータは、True とします。False が設定された場合、無視され「False」が返ります。

この関数を使うと、アクティビティモニタの出力内容をコントロールできるので、欲しい情報だけを的確に採取することができるようになります。

#### 例: ゲートウェイログ出力を特定のタスク実行中にだけ限定する

例として、あるタスクに関してだけ、ゲートウェイのログを調査したいときがあるとします。「ロギング」ダイアログ や MAGIC.INIの設定でゲートウェイログを出力するようにしておくと、アプリケーションの最初から最後まですべ てゲートウェイのログがアクティビティモニタやログファイルに記録されるようになり、複雑なロジックのアプリ ケーションでは、膨大なログ出力となることがあります。

しかし、本当にゲートウェイログを採取したいタスクが特定されている場合には、次のようにして、そのタスクの 実行期間中だけログを記録するようにできます。

- 「メインプログラム」のタスク前処理で、Logging ('FALSE'LOG, 'ALL')を実行し、すべてのログ出力を抑制 します。
- ログを記録したいタスクのタスク前処理で、Logging ('TRUE'LOG, 'Gateway')、および Logging('TRUE'LOG, 'MicrosoftSQLServer=D')を「アクション」コマンドで実行します。(DBMS が SQL Server で、ログのレベルがD(開発者)の場合)
- タスク後処理で、再度 Logging ('FALSE'LOG, 'ALL')を実行し、ログ出力を抑制します。

## 10.2. FlwMtr 関数

実行の記録のために、アクティビティモニタに任意のメッセージを記録しておくことができます。これには FltMtr 関数を使います。

- **構文:** FlwMtr (文字列, 論理ブレイク)
- パラメータ: 文字列 (文字型/Unicode 型)… アクティビティモニタに表示させる文字列
  - 論理ブレイク(論理型)… V1Plus では無効です。

**戻り値**: 常に False が返ります。

FlwMtr 関数を実行すると、指定した「文字列」がアクティビティモニタに記録されます。アクティビティモニタには、 他のログ (タスク起動・終了、フロー、DBMS 活動) なども記録されるので、例えば「このコマンドはどのタイミン グで実行されるのか?その時の項目の値は何か?」などを調べたい場合に、デバッグ用に FlwMtr 関数を使っ て記録をすることができます。



リファレンスマニュアルでは、第2パラメータ「論理ブレイク」として、'TRUE'LOG の場合にデバッグ ブレークがかかるとありますが、V1Plus では無効になっています。

# 10.3. 項目の表示プログラム

uniPaaSのプログラムを実行している時、特定の時点での変数の値を見てみたい、と思うことがあります。 デバッガが利用できる場合には、ブレークを入れて、「項目」画面を表示させることにより、その時点で有効な項 目の値を見ることができます。しかし、(1)ブレークを入れたくない場合、(2)デバッガが利用できない場合、(3) ファイルなどに記録をとっておきたい場合、などでは別の方法を考える必要があります。 このような場合には、ちょっとした uniPaaS プログラムの作成が必要になりますが、Var 関数を使って実現する ことができます。

プログラム「項目出力」は次のように作ります。

タスク特性	タスク名:項目出力 タスクタイプ: バッチ 初期モード: 修正 タスク終了条件: Counter(0) >= 'A'VAR
入出力ファイル	名前: Out メディア: F=ファイル アクセス: A=追加 書式: L=ライン 式: 'var.log'
データビュー	変数 A「this counter」数值型、書式 8
ロジック/レコード後処理	項目更新 A = Counter(0) フォーム出力 O=出力、フォーム: 4 (out line)、ファイル: 1 (Out)
フォーム/4 out line	クラス: 1 インターフェースタイプ: T=テキスト形式 幅: 230、高さ: 1 項目1: (0,0) - (30,1)、型: U=Unicode、書式: 30、データ= VarName(a) 項目2: (31,0) - (200,1)、型: U=Unicode、書式: 200、 データ: CASE (VarAttr (A), 'A', VarCurr(A), 'N', Str(VarCurr(A),'N16.2'), 'L', IF(VarCurr(A),'TRUE','FALSE'), 'D', DStr(VarCurr(A), 'TRUE','FALSE'), 'D', DStr(VarCurr(A), 'YY/MM/DD'), 'T', TStr(VarCurr(A), 'HH:MM:SS'), 'U', VarCurr(A), '(type:' & VarAttr(A) & ')')

Var 関数を簡単に説明すると、VarAttr は項目の型を1文字で返します (文字型 → 'A'、数値型 → 'N' など)。 また、VarCurr は、項目の現在の値を返します。値の型は、パラメータで指定した項目の型そのものです。

「項目出力」プログラムを簡単に説明すると、このプログラムでは、VarCurr(A)の値をファイルに出力します。 フォームの項目2がその値ですが、この式が CASEを使った複雑なものになっているのは、VarAttr (A)のタイ プが文字型とは限らないので、文字型に変換するために、それぞれの型に合った組み込み関数と書式とを使っ ているためで、やりたいことは、VarCurr(A)の値を取得することです。

このバッチタスクはループを作りますが、その終了条件は Counter(0) >= 'A'VAR となっています。

これだと、「このプログラムは、自分のタスクの変数 A の値を出力するだけではないか? 変数 A の値は数値型

第10章 プログラミングの小技

で Counter(0)に代入されているが、終了条件が Counter(0) >= 'A'VAR とはどういうことだ?」と思われるかもし れません。この謎を解く鍵は、次の二つの Magic ルールにあります。

- 1. VARリテラルの値は、実際には数値です。例えば、'A'VAR = 1、'B'VAR = 2、… というように、'A' を起点とす る連続整数値となります。
- 2. 実行時には、VARリテラルの値は、メインプログラムから現在実行中のタスクまでのすべてのタスクで定義 されているデータ項目を積み重ねて、最初(メインプログラムの最初の項目)を1と数えた連続数値となってい る。

データビュー ロジック フォーム

1は簡単に理解できると思いますが、2は少し説明が必要です。

例として、タスクA、B、C があり、そ 🌏 🗛 🛛 13 - A れぞれ、2個、2個、4個の項目が データビューに定義されているとし ます。

そして、タスクAがBを、BがCを 呼び出したとします。

この状態で、上記の「項目出力」タス クを呼び出します。

	1 2 3	¥=メインソー 0 ∀=変数 1 ∀=変数 2	<b>メインソース未定義</b> 今日の日付 今の時刻	<b>インデゥ0</b> D=日付####/## T=時刻HH:MM:S	代入I 代入2	Date() Time()	
ĺ	<b>@                                    </b>	4 – B					X
	データビュ	ユニ ロジック	フォーム				
	1	■=>{>>+ 1	NA	インデ [*] ッ1			<u>_</u>
	2	じ=カラム I C=カラム 2	A	N=致1但5 A=文字10			

🔵 ዓスク	15 -	С				X
データヒ	[	ロジュ	ック	フォーム		
1	ii-	ッインソー	6	Products	{)デッ1	~
2	C	= カラム	1	Product ID	N=裝(直N10	
3	C	= カラム	2	ProductName	U=Unic40	
4	C	= カラム	4	CategoryID	N=娄y值N10	
5	C	= カラム	6	UnitPrice	N=数値10.3	~

このとき、VARリテラルの値は、右 図のようになっています。

これからわかるように、「項目出力」 タスクの 'A'VAR は、この状態では 9になっています。

タスク	変数名	VAR リテラル値
A	今日の日付 今の時刻	1 2
В	N A	3 4
С	ProductID ProductName CategoryID UnitPrice	5 6 7 8
項目出力	this counter	9

VARリテラルの値についてのルールを理解すれば、終了条件 Counter(0) >= 'A'VAR は、この場合 Counter(0) >= 9 と書いたのと同じで、Counter(0) が 1 ~ 8 まで、8 回のループが走ることがわかります。

レコード後処理では、Counter(0)の値を A に代入して、VarCurr(A)を出力しますから、ループで VarCurr(1) ~ VarCurr(8) までについて出力がされます。このようにして、「項目出力」タスクは、その時点で有効な項目をすべ て出力できるようになります。

ファイルに出力された結果は、右図	変数項目. 今日の日付	10/11/02
のようになります。	変数項目. 今の時刻	18:37:36
	NA. N	1.00
	NA. A	aaabbc
	Products.ProductID	6.00
	Products. ProductName	Grandma's Boysenberry Spread
	Products.CategoryID	2.00
	Products.UnitPrice	25.00



この項目表示プログラムは、コンポーネント化しておけば、デバッグしているアプリケーションから 簡単に呼び出すことができるようになります。

# 10.4. コマンドラインリクエスタのログ読み込みプログラム

uniPaaSのコマンドラインリクエスタ(Mgrqcmdl.exe)のログ出力機能により、実行されたリクエストについての情報を一覧で取得することができます(4.1.4「個々のリクエストについて表示させるには?」参照)。 その出力フォーマットは、次のようなものです。

C:¥Program Files¥uniPaaS¥Enterprise Server V1Plus>mgrqcmdl -query=log						
	Log	of requests (MyServer/5215, 10:11:21)				
# Request Id	Status	Start Elapsed Completion Codes Time (sec.) (Mri, Runtime, Dbms)				
1   22    Program : Priority : Client : 00	DONE ″NA_MAIN 0 000000EF4	16/11 09:39:42   0   -OK (0) 0 47 in sub (update fail)" ("subtask-link") #2B255_D (pid 5656), EnterpriseServer : MyServer/	1501			
= = = = = = = = = = = = = = = = = = =	= = = = DONE "NA_MAIN 0 000000FF4	= = = = = = = = = = = = = = = = = = =	= = = = =			
= = = = = = = = = = = = = = = = = = =	= = = = = 下省略)	= = = = = = = = = = = = = = = = = = =	= = = = =			

この出力はテキストファイルに格納されますが、データベース化することにより、解析が容易になります。例えば、実行時間 (Elapsed sec.) が長い順にリクエストを表示させたいとか、あるいはクライアント(Client) 別にリクエストをグループ化して調査したい、というような場合、テキストファイルのままでは難しいですが、データベース化することで簡単になります。

このようなデータベース化プログラムは、uniPaaS で簡単に作成することができ、基本的にはテキストファイル読 み込み → 行の解析 → レコード作成、という処理になります。

ログファイルは、次の5行が1リクエストに対応していて、このパターンの繰り返しとなっています。

1 | 22 |DONE | 16/11 09:39:42 | 0 | -OK (0) 0 47 Program : "NA_MAIN in sub (update fail)" ("subtask-link") Priority : 0 Client : 00000000EF42B255_D (pid 5656), EnterpriseServer : MyServer/1501

それぞれの項目の意味は次の通りです。

<#> | 〈リクエスト ID〉 |〈ステータス〉| 〈日付〉〈時刻〉| 〈処理時間〉| 〈結果ステータス〉 Program : "〈プログラム名〉"("〈アプリケーション名〉") Priority : 〈優先度〉 Client : 〈クライアント ID〉 (pid 〈プロセス ID〉), EnterpriseServer : 〈サーバホスト名〉/〈サーバポート番号〉 各項目は固定長ではないので、各行から InStr、StrTok などの関数を使って項目値を切り出すようになります。 uniPaaS プログラムとしては簡単なプログラムですので、詳しいことは実際のプログラムを見てください。 次の図は、このログ読み込みプログラムを使って読み込んだ例です。このように uniPaaS でデータベース化して しまえば、簡単なものはオンラインプログラムの実行時オプション(位置づけ、範囲付、ソート)で並び替えがで きるし、少し複雑なロジックを必要とするようなものでも、uniPaaS プログラムを作って対応することができるよう になります。また、DataViewToText/DataViewToXML 関数などを使って、CVS ファイルや XML ファイルに出力 して、Excel などで分析することもできます。

	.og_Q	UERY_LO	G							and the second s		×
L	ine No	Reques	. Log Status	Recieve Time Elapsed	End Time	Log Reason	Log Reason Code	Log DBMS Code	Program Name	Client	Server Name	*
	61	1	DONE	14:30:34	1 14:30:35	-OK	0	0	rc	00000000EF42B255_D	areyomi	
	56	2	DONE	14:30:35	0 14:30:35	-OK	0	0	rc	00000000EF42B255_D	areyomi	
	51	3	DONE	14:30:35	0 14:30:35	-OK	0	60	rc	00000000EF42B255_D	areyomi	
	46	4	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	41	5	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	36	6	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	31	7	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	26	8	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	21	9	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	16	10	DONE	14:30:35	0 14:30:35	-ОК	0	0	[Caching]	00000000EF42B255_D	areyomi	
	11	11	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	6	12	DONE	14:30:35	0 14:30:35	-OK	0	0	[Caching]	00000000EF42B255_D	areyomi	
	1	13	DONE	14:30:37	0 14:30:37	-OK	0	60	rc	00000000EF42B255_D	areyomi	~
										照会		挿入

# 10.5. アクティビティモニタ出力の読み込みプログラム

アクティビティモニタの出力(2.1「アクティビティモニタとは何ですか?」以下を参照)も、テキストファイルから データベース化することにより、分析が容易になります。このような読み込みプログラムも、uniPaaSで簡単に作 成することができます。

アクティビティモニタの出力例を以下に示します。

<7819595268872 > 09:39:27.422	->	>情報 >> IO file 'SUPPORT¥html_std.jpn' cannot be OPENED
<15639190537744> 09:39:33.575	->	> Calling task 'rc1' ('subtask-link') : Session Counter #4 (Request ID #3)
<15639190537744> 09:39:33.579	->	>情報 >> IO file 'SUPPORT¥html_std.jpn' cannot be OPENED
<15639190537744> 09:39:33.610	-	>>開始 ロート・ バッチ タスク - 'メインプログラム (subtask-link)' 照会 モート・
<15639190537744> 09:39:33.612	-	終了 タスクのロード
<15639190537744> 09:39:33.614	-	レコート"読込
<15639190537744> 09:39:33.675	-	>>開始 ロードリッチクライアント タスク ー 'NA_MAIN in sub (update fail)' 修正 モード
<15639190537744> 09:39:33.676	-	終了 タスクのロード
<15639190537744> 09:39:33.679	-	レコート"読込
<15639190537744> 09:39:33.682	-	新規の遅延トランザクション書込 2
<15639190537744> 09:39:33.683	-	開始 レコード前
<15639190537744> 09:39:33.684	-	終了 レコード前
<15639190537744> 09:39:33.730	-	>> Completed Request ID #3 (0.16 seconds)
<15639190537744> 09:39:33.731	-	>>
<15639190537744> 09:39:33.789	-	>> 情報 >> Request's Session Counter #0 <> Server's Session Counter #4
<15639190537744> 09:39:34.447	-	>> イベント処理中[コンテキストがフォーカス喪失(L)]コントロール名 : [ 無効 ]
<15639190537744> 09:39:34.462	-	>> Calling task 'rc1' ('subtask-link') : Session Counter #5 (Request ID #13)
<15639190537744> 09:39:34.001(c	) –	Starts Record Prefix
<15639190537744> 09:39:34.004(c	) –	Flow - (Step Forward)
<15639190537744> 09:39:34.475	-	フロー ー コール S=サブタスク: NA_MAIN(通常モード 前方)
<15639190537744> 09:39:34.507	-	4,84627 ms7_ini(): >>>>>
<15639190537744> 09:39:34.509	-	,84627 ms7_ini(): MicrosoftSQLServer , Version uniPaaS 1.8 SP1a PT1-0 19-Jan-2010
<15639190537744> 09:39:34.510	-	,84627 Version uniPaaS 1.8 SP1a PT1, Log = mgmonitor.log, Level = Support/QA, Sync = Reopen,
<15639190537744> 09:39:34.512	-	,84627

この出力は、基本的に1行1レコードに対応し、フォーマットは次のようなものです。

<コンテキスト ID> <時刻> <メッセージ>

1行づつ読み込み、このフォーマットに従ってレコードを作成していく、というようなプログラムになります。 ただし、アクティビティモニタの出力はばらつきがあり、データベースとして標準化するために細かな加工が必 要になります。

- コンテキストID: コンテキストID は「<」と「>」とで囲まれた数値ですが、桁数はまちまちであり、可変長として切り出す必要があります。また、「-1」というものもあります。
- 時刻: ミリ秒(小数点以下3桁)の単位まで表示され、12桁固定です。
- メッセージ:この部分は、メッセージの内容により形式がかなり違います。
- 情報、タスクやイベントの開始/終了、フローの実行などは、「-」記号の後に、複数の空白文字でインデントされています。解析する上でインデントは普通要らないので、「-」記号と合わせて、LTrim で削除してしまいましょう。
- ゲートウェイの出力では、「-」記号の後にもう一つの時刻が記録されています。例えば、

<15639190537744> 09:39:34.507 - **4,84627** ms7_ini(): >>>>>

の「4,84627」の部分です。これはこれで有用な情報なのですが、メッセージとは切り離して別項目として格納した方が良いでしょうから、切り離します。

● リッチクライアントタスク実行中には、クライアント側の活動ログも併せて記録されます。クライアント側の ログは、次の例のように、<時刻>の直後に「(c)」という記号で区別されています。 <15639190537744> 09:39:34.001(c) - Starts Record Prefix

これもまた有用な情報なので、メッセージとは切り離して、別項目として格納します。

 データベースゲートウェイの出力で、改行コードがメッセージの中に入っているものがあります。例えば、 次のようなものがあります。

<15639190537744> 09:39:35.672 - ,85782 ms7_esqlc_fil_exist(): select id from MAGICDB..sysobjects where name = 'NA_LINK' AND uid = user_id(") AND (type = 'U' OR type = 'V' OR type = 'S')

このような場合には、2行以上を1行に連結してやる必要があります。連結の必要があるか否かは、行頭 に「<…>」の形式のコンテキストID があるかどうかをチェックして確認します。コンテキストID が認識でき ない行は、前行からの継続行として、前行のデータに連結します。

#### 以下に、このプログラムでデータベース化したログのイメージを示します。

📰 Bro	owse - LOGLINE			
LINE	Ct×ID	TIME	MSEC	A
1	7819595268872	09:39:27.422		>> 情報 >> IO file 'SUPPORT¥html_std.jpn' cannot be OPENED
	2 15639190537744	09:39:33.575		>> Calling task 'rc1' ('subtask-link') : Session Counter #4 (Request ID #3)
	3 15639190537744	09:39:33.579		>> 情報 >> IO file 'SUPPORT¥html_std.jpn' cannot be OPENED
	4 15639190537744	09:39:33.610		>>開始 ロード バッチ タスク - 'メインプログラム (subtask-link)' 照会 モード
	5 15639190537744	09:39:33.612		終了 奴/のロート*
	6 15639190537744	09:39:33.614		レコート「読込
	7 15639190537744	09:39:33.675		>>開始 ロード リッチクライアント タスク - 'NA_MAIN in sub (update fail)' 修正 モード
	8 15639190537744	09:39:33.676		終了 奴功のロート*
	9 15639190537744	09:39:33.679		Jコード読込
	10 15639190537744	09:39:33.682		新規の遅延トランザクション書込 2
	11 15639190537744	09:39:33.683		開始 レコード前
	12 15639190537744	09:39:33.684		終了 レコード前
	13 15639190537744	09:39:33.730		>> Completed Request ID #3 (0.16 seconds)
	14 15639190537744	09:39:33.731		»
	15 15639190537744	09:39:33.789		>> 情報 >> Request's Session Counter #0 <> Server's Session Counter #4
	16 15639190537744	09:39:34.447		>> イベント処理中 [コンテキストがフォーカス喪失(L)] コントロール名: [ 無効 ]
	17 15639190537744	09:39:34.462		>> Calling task 'rc1' ('subtask-link') : Session Counter #5 (Request ID #13)
	18 15639190537744	09:39:34.001		C Starts Record Prefix
	19 15639190537744	09:39:34.004		C Flow - (Step Forward)
	20 15639190537744	09:39:34.475		フロー - コール S=サブタスク: NA_MAIN (通常モード 前方)
	21 15639190537744	09:39:34.507	4,84627	ms7_ini(): >>>>>
	22 15639190537744	09:39:34.509	,84627	ms7_ini(): MicrosoftSQLServer , Version uniPaaS 1.8 SP1a PT1-0 19-Jan-2010
	23 15639190537744	09:39:34.510	,84627	Version uniPaaS 1.8 SP1a PT1, Log = mgmonitor.log, Level = Support/QA, Sync = Reop
	24 15639190537744	09:39:34.512	,84627	ms7_ini(): <<<<< retcode = 36 🗸 🗸
→ 情   >> 情	辑 >> IO file 'SUPPC	RT¥html_std.jpn' ≀	cannot be OF	ENED
				修正 挿入 /

このようにデータベース化することにより、例えば、次のようなことができるようになります。

- コンテキスト ID 15639190537744 の活動だけを表示させたい: CtxID カラムで範囲付けを行います。
- テーブル NA_MAIN の活動だけを表示させたい:「InStr (G,'NA_MAIN') > 0」という式で範囲付けします。
- タスクの実行順序を調べたい:「>>開始 ロード」で始まる行のみ表示させます。



トラブル シューティング ツール Copyright © 2010 Magic Software Japan K.K., All rights reserved.

Magic uniPaaS V1Plus

第2版 2011年2月8日

発行〒151-0053 東京都渋谷区代々木三丁目二十五番地三号<br/>あいおいニッセイ同和損保新宿ビル 14 階<br/>マジック ソフトウェア・ジャパン (株)<br/>http://www.magicsoftware.co.jp/