株式会社ミガロ.

RAD事業部 技術支援課

# [Delphi/400] Enterprise Connectorsを 利用したクラウド連携テクニック

1. はじめに

- 2. Enterprise Connectors とは
- 3. Enterprise Connectors のインストール
- 4. Enterprise Connectors の開発方法
- 5. クラウドサービスへの接続・活用方法
- 5-1. Twitter と連携した情報発信と解析
- 5-2. Google スプレッドシートへのデータ出力
- 5-3. Google Drive でのデータ検索
- 6. まとめ

## 1. はじめに

基幹系の業務システムは長らく、社内 上のサーバーにデータを蓄積するオンプ レミス(自社運用)で稼働してきた。 Delphi/400のシステムでも、IBM iを 中心にオンプレミス環境で稼働している ことが多い。

従来はすべてのシステムをオンプレ ミスで稼働させるのが一般的だったが、 近年は Web を介して稼働するさまざま なクラウドサービスが台頭しており、そ れらを業務で活用することも増えてい る。オンプレミスのシステムとクラウド サービスは別々に稼働していることが多 いのが現状であり、これらの連携が課題 になることもあるだろう。このようなオ ンプレミスとクラウドサービスとの連携 でも、Delphi/400が活用できる。

本稿では、Delphi/400からクラウド サービスへのアクセス用に用意された Enterprise Connectors を利用して、 IBM i のデータとクラウドサービスを連 携させる方法について紹介する。

## 2. Enterprise Connectorsとは

Enterprise Connectors とは、エンバ カデロ・テクノロジーズ社が CData Software 社との提携によって、2017 年 から提供を開始したコンポーネント群 で、Delphi/400 に搭載されたデータベー スエンジン FireDAC を使用してクラウ ドサービスに接続する。

各クラウドサービスには、それぞれの ベンダーが提供する Web API があり、 それを使用することで Delphi/400 等の 各種クライアントからクラウドサービス に接続できた。しかし「クラウドサービ スごとに接続する API の仕様が異なり、 個別に API の仕様を調査の上、実装し なければいけない」「API 側の仕様の アップデート間隔が短いことが多く、仕 様変更の都度、個別に修正を実装しなけ ればならない」など、課題が多いのが現 状である。

また API の基本設計自体が変わる、 たとえばデータ形式そのものが XML か 略歴 1985年12月6日生まれ 2009年3月甲南大学経営学部卒業 2009年4月株式会社ミガロ、入社 2009年4月システム事業部配属 2019年4月 RAD事業部配属

現在の仕事内容 Delphi/400でのシステム開発や保守作業の経 験を経て、現在はサポート業務を担当している。

ら JSON に変更された例も存在する。

Enterprise Connectors は、こういっ たベンダー側の API 仕様をラッピング し、FireDAC の共通インターフェース で活用できるようにした。これにより各 種クラウドサービスに対して、IBM iや オープン系の DB へのアクセスと同様 に、SQL やストアドプロシージャを使 用してアクセスできる。【図 1】

Enterprise Connectors のサブスクリ プションには、主要な約 80 種類のクラ ウドサービスに接続可能な「Enterprise」 版と、約 150 種類のクラウドサービス に接続可能な「Enterprise Plus」版が 存在する。

本稿ではこれらの中から代表的なも のとして、業務に限らず広く使われてい る Twitter および Google スプレッド シートを取り上げ、Delphi/400と連携 可能な利用例について紹介する。

これらの連携を使用すれば、たとえば 「IBM i 上の売上データをもとに Google Drive 上に Google スプレッドシートを 作成する」「IBM i 上の余剰在庫情報を

	*	Enterprise 版で約 Enterprise Plus 約 (2019年10月現在	580種類が、 仮で約150種類が E)	利用可能
IBM i $ \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c} \begin{array}{c}$	サービス	Salesforce & Force.com Microsoft Dynamics NAV Microsoft SharePoint Apache Spark SQL Microsoft SharePoint Salesforce & Force.com Microsoft Dynamics NAV FreshBooks Accounting Microsoft SharePoint Salesforce & Force.com Microsoft Dynamics NAV Microsoft SharePoint Microsoft SharePoint Micro	<ul> <li>Microsoft Dynamics CRM</li> <li>Microsoft Dynamics AX</li> <li>Apache Cassandra</li> <li>NetSuite CRM &amp; ERP</li> <li>NetSuite CRM &amp; ERP</li> <li>Xero Accounting</li> <li>SAP NetWeaver</li> <li>Microsoft Excel</li> <li>Facebook</li> </ul>	Microsoft Dynamics GP MongoDB Rei Apache HBase Sege Sage 50 Accounting E Exact Online E OData Services SharePoint Excel Servi- LDAP Directory Services
MS SQL Server・ Oracleなど各種RDBMS		Twitter	Facebook	LDAP Directory Service
Enterprise Connectorsの取得		Anazori Dynamous		Coogle Analytics
e Enterprise Connectorsの取得 キー入力ページ ⇒ https://reg.code	egear.com/srs6/p Follow Us:	promotion.jsp?prom	noId=561	Google Analytics
e Enterprise Connectorsの取得 キー入カページ ⇒ https://reg.code embarcadero company products	egear.com/srs6/p Follow Us:	promotion.jsp?prom	noId=561	Coogle Analytics
Enterprise Connectorsの取得 キー入力ページ ⇒ https://reg.code COMPANY PRODUCTS RAD Studio, Delphi or C++Builder / Subscription - Claim your Enterpris here これはプロモーションサービスを入手するためのページです。	egear.com/srs6/p Follow Us: C C C SOLUTIONS Architect or Enter e Connectors (CE ソフトウェアを登録しようとし	promotion.jsp?prom RESOURCES prise license holders Data FireDAC Profes	oold=561 بوهرداری News & EVENTS s with current Upda ssional) Subscriptio	Coogle Analytics
Enterprise Connectorsの取得 キー入力ページ ⇒ https://reg.code Company PRODUCTS RAD Studio, Delphi or C++Builder/ Subscription - Claim your Enterpris here これはプロモーションサービスを入手するためのページです.	egear.com/srs6/p Follow Us: C C C SOLUTIONS Architect or Enter e Connectors (CC ソフトウェアを登録しようとし	promotion.jsp?prom RESOURCES prise license holders Data FireDAC Profes してこのページを開いた場合には、こ lphi を購入した際の Jアルナンバーを入	ioId=561 NEWS & EVENTS s with current Upda ssional) Subscriptio	Coogle Analytics

もとに、Twitter にキャンペーン情報を ツイートする」といった使い方ができる。 なおEnterprise Connectors は、 Delphi/400 Version 10.2 Tokyo からの 対応となっており、本稿のサンプルでも 同バージョンを使用している。また本稿 で扱うそれぞれのコネクターは、 「Enterprise」版のサブスクリプション で利用可能である。

## 3. Enterprise Connectorsの インストール

Enterprise Connectors は、Delphi/ 400 のオプションのライセンス製品であ るが、アップデートサブスクリプション が有効な Delphi/400 10.2 Tokyo をご 利用であれば「Enterprise」版を無償で 取得できる。

まず、エンバカデロ・テクノロジーズ 社のサイト (https://reg.codegear.com/ srs6/promotion.jsp?promoId = 561) にアクセスし、Delphi 購入時のシリア ルナンバーを入力して Product Key を 取得する。【図 2】

次に CData Software 社のサイト (https://www.cdata.com/firedac/ download/) にアクセスし、必要なコネ クターを選択すると【図3】のような画 面が表示される。それぞれのコネクター は独立しており、コネクターごとに専用 のインストーラと、それによってインス トールされる専用のコンポーネントが個 別に存在する。

先ほどの Product Key とそれを取得 するために使用したメールアドレスを入 力すると、対象のコネクターのインス トーラがダウンロードされる。

インストールが完了すると、ヘルプド キュメントがスタートメニューに登録さ れる。そのコネクター関連の各コマンド や使用方法が英語で記載されているの で、開発時の大きな助けになるだろう。 【図 4】

### 4. Enterprise Connectorsの 開発方法

本章では、各種コネクターの中でもシ ンプルで扱いやすい「CData CSV FireDAC Components」を使用して、 CSV ファイルにアクセスする方法を紹 介する。

CSV といえば、通常は TStringList などで読み書きすることが多いが、 Enterprise Connectors の使用によっ て、CSV データに対して、一般的な DB アクセスと同様に SQL を使用した データの抽出や更新が可能になるメリッ トがある。

それでは、具体的なサンプル作成を通 じて開発方法を紹介する。今回のサンプ ルでは「CSV ファイルを読み込み、 TDBGrid に表示」「IBM i から取得し たデータをもとに CSV ファイルにレ コードを追加」の2つの機能を実現する。 Delphi/400 10.2 Tokyo を起動した ら、アプリケーションを新規作成し、新

しいフォームの上にコンポーネントを配 置する。 まず Enterprise Connectors で CSV

に接続してデータを読み書きするための コンポーネントとして、「CData CSV FireDAC Components」をインストー ルした際に追加されるコネクター 「TFDPhysCDataCSVDriverLink」 を 配置する。

これは IBM iへの接続における TFDPhysCO400DriverLink に相当し、 Enterprise Connectors はコネクターご とのドライバリンクコンポーネントを定 義するのがポイントである。また通常の FireDAC 接続と同様に TFDConnection・ TFDQuery と、取得結果を画面に表示す る TDataSource・TDBGrid を配置する。

次に IBM i に接続してデータを取得 するコンポーネント群として、 TFDConnection・TFDQuery および、 IBM i に接続するためのコネクターと なる TFDPhysCO400DriverLink を配 置する。

あとは処理実行のために TEdit や TButton などを配置した結果、【図5】 のような画面になる。

そして TFDConnection をダブルク リックして、FireDAC 接続エディタを 開き、ドライバ ID に「CDataCSV」を 選択する。すると【図 6】のように、接 続時の各種パラメータが自動でセットさ れる。

ローカルの CSV ファイルに接続する 場合は、「URI」パラメータに対象 CSV のフォルダを指定する。また「UseRow Numbers」や「RTK」といったパラメー タを【図 6】のように設定する。

次に、接続する CSV を準備する。本 稿のサンプルでは、【図7】のようなレ イアウトで「WORKCSV.csv」という サンプル CSV を作成した。CSV のファ イル名は、拡張子を除いた「WORKCSV」 部分がテーブル名に、1 行目のカンマ区 切りテキストがそのままフィールド ID に、2 行目以降のカンマ区切りテキスト がレコードの内容となる。

#### CSV ファイルを読み込み、TDBGrid に表示

画面設計が完了したら、【ソース1】 のようにロジックを記述する。プログラ ムを実行して「① SQL SELECT」ボタ ンを押下することで、CSV の内容が DBとして明細に表示される。【図8】の TDBGrid の内容は、単純な全件取得の 結果であるが、一部フィールドだけの取 得や、WHERE 句を使用した絞り込み も可能である。

FireDAC接続エディタの設定で、 「UseRowNumbers」をTrueに指定し て書き込みを有効にした場合、行番号で 各レコードをユニークキーとするための 「RowNumber」というフィールドも追 加される。【図8】では明示的に表示さ れたままにしているが、通常は列ごと非 表示に設定しても問題ない。

(2) IBM i から取得したデータをもとに CSV ファイルにレコードを追加

次に、【ソース 2】のようにロジック を記述する。顧客コードに値を入力して 「②登録」ボタンを押すと、IBM i から 条件に合致するデータを取得し、その結 果 を SQL の INSERT 文 を 使 用 し て CSV に行を追加できる。【図 9】

## 5. クラウドサービスへ の接続・活用方法

前章では、Enterprise Connectors の 基本的な開発について紹介したが、ここ では具体的なクラウドサービスの連携例 として、業務に限らず広く一般でも使用 される Twitter および Google Apps (Google スプレッドシートおよび Google Drive) について紹介する。

CData Software – Thank you for your interest in the Inst then click download to download the	DOWNLOADS agram FireDAC Components. Please fill in the required (*) contact information below and setup.
Download Instagram FireDA	C Components たほどProduct Keyの取得に
Company Email* :	使用したメールアドレス
Product Key*:	たほど取得したProduct Key (複数コネクターでもKeyは同一)
By including your optional email addre products and services. Your informatio our full Privacy Policy. please click her By downloading and installing this pro 初回のコネ	ess above, you agree to receive periodic communication from CData Software regarding our n will be kept entirely confidential and used only by authorized members of our staff. For e. duct you agree to comply with the product End User License Agreement. クターのみ、インストール時に
氏名とメーク	ルアドレスの入力が必要 Enal Country ysade@migaro.co.jp *
	Please send me product information including special offers and promotions.
	www.cdata.com
Enterprise Conne	www.cdata.com <back next=""> Cancel ectorsのヘルプドキュメント</back>
4 Enterprise Conne コネクターごとに信	www.cdst.com www.cdst.com ectorsのヘルプドキュメント 使用できるメソッドやロジックの記法が異なるため、
4 Enterprise Conne コネクターごとに使 それぞれのヘルフ	www.cdits.con www.cdits.con ectorsのヘルプドキュメント 使用できるメソッドやロジックの記法が異なるため、 パドキュメントが存在する
4 Enterprise Conne コネクターごとに信 それぞれのヘルフ く → C ① ファイル I C:	www.cdd.s.con ectorsのヘルプドキュメント 使用できるメソッドやロジックの記法が異なるため、 パドキュメントが存在する /Program%20Files/CData/CData%20FireDAC%20Components%20for%20Twitter/help/help.htm Program%20Files/CData/CData%20FireDAC%20Components%20for%20Twitter/help/help.htm CData FireDAC Components for Twitter 2013

#### 5-1. Twitterと連携した情報発信と 解析

最初に、有名人・著名人も数多く利用 するソーシャルメディアの1つで、 Google 検索でもヒットしないような口 コミをリアルタイムで調べられる Twitterの接続、利用方法を紹介する。

#### Twitter API のアプリケーション開 発準備

ま ず、Twitter の 開 発 者 サ イ ト (https://dev.twitter.com/apps) から、 新しい Twitter Web API を作成する。 開発者サイトにログインしたら、「Create an app」ボタンを押し【図 10】、作成す るアプリケーションのタイトルや使用目 的を英語で入力する。【図 11】

入力が完了したら、Twitter アプリ ケーション開発に必要な API キーとア クセストークンを入手できる。【図 12】 【図 13】

なお開発者サイトを利用するにあ たっては、Twitterの開発者アカウント の作成が必要である。本稿では作成手順 については割愛するが、Google 等で 「Twitter 開発者アカウント」と検索す れば、その時点での最新情報が参照でき るだろう。

#### (2) Twitter API を使用したツイート送 信方法

APIを作成したら、次は Delphi/400 側の開発を行う。ここでは、「CData Twitter FireDAC Components」を利 用する。Delphi/400 10.2 Tokyo を起動 したら、アプリケーションを新規作成し、 新しいフォームに【図 14】のようにコ ンポーネントを配置する。

前章の CSV 接続と同様、Delphi/400 で IBM i に接続してデータを取得する コンポーネント群と、Enterprise Connectors で Twitter に接続するコン ポーネント群をそれぞれ配置する。なお、 Twitter で使用するコネクターは、 「TFDPhysCDataTwitterDriverLink」 である。

前章のCSV接続と同じように、 TFDConnectionをダブルクリックして FireDAC接続エディタを開き、ドライ バIDに「CDataTwitter」を選択すると、 【図15】のように接続時の各種パラメー タが自動でセットされる。

ここでセットが必要なパラメータは、 接続目的にあわせて【図 16】のように 設定する。また「②選択行の商品につい てツイートする」ボタン押下時の処理を、 【ソース3】のように記述する。TFDQuery を使用して、テーブル「Tweets」に SQL でレコードを追加するだけで、ツ イートが送信できる。

Delphi/400 で IBM i からデータを取 得し、【図 17】のように TDBGrid に表 示した状態から、「②選択行の商品につ いてツイートする」ボタンを押下すると、 【図 18】のようにツイートが送信される。 送信されたツイートにはユニークキーと なる ID(数字 18 ~ 19 桁)が文字型で 採番されており、ブラウザの URL で確 認できる。

送信時の SQL により、「In\_Reply\_ To\_Status\_Id」フィールドに別のツイー トの ID をセットすると、そのツイート へのリプライになる。またツイートの本 文中に別のツイートの URL「https:// twitter.com/XXX/status/(ツイート の ID)」を記載すると、そのツイートに 対する引用リツイートになる。

本稿の例では SQL の INSERT 文で ツイートを送信したが、逆に SQL の DELETE 文を発行することで、自分が 送信したツイートを削除することも可能 である。その際には、対象ツイートの ID を WHERE 句に指定する。

なお、それぞれのツイートは FireDAC 接続時に認証を受けたユーザーのアカウ ントで発信されるため、当然のことなが ら他人のアカウントになりすますことは できない。

# (3) Twitter API を使用した各種データの参照方法

次に、Twitter APIを使用して各種 テーブルや、キーワードを指定したツ イートを検索するアプリケーションを作 成する。

Delphi/400 10.2 Tokyo でアプリケー ションを新規作成し、新しいフォームに 【図 19】のようにコンポーネントを配置 する。【ソース 4】のようにロジックを 記述し、プログラムを実行する。「①タ イムラインの表示」ボタンを押下すると、 「SELECT \* FROM Tweets」 という SQL が発行され、自分(ログイン中の ユーザー)と自分がフォローしている ユーザーのツイートが一覧で表示され る。【図 20】

「②フォローしているユーザーを表示」「③自分がいいねを押した投稿を表示」「④フォロワーの一覧を表示」の各 ボタン押下時の結果についても、【図 20】で示したように参照できるので、ぜ ひ一度試していただきたい。

さて、ここで使用したテーブル 「Tweets」は、タイムラインを表示する だけのテーブルではない。キーワードを 指定して検索する機能も持っている。

【ソース5】のようにロジックを記述 する。WHERE句にフィールド「Search Terms」を指定することで、指定され たキーワードを含む直近の6~9日以内 のツイートを検索できる。【ソース5】 のロジックでは、影響度が高いツイート を絞り込むため、「リツイート数が一定 以上」「いいね数が一定以上」「日本語の ツイートに限定」という条件も付けられ るようにしている。

【図 21】 の 明 細 は「Enterprise Connectors」というキーワードで、 Twitter 検索を行った結果である。

検索結果は、レコードの作成日時が新 しい順(ツイートが新しい順、最近フォ ローされた順など)や、Twitter が独自 に判断した影響度順で表示されており、 ORDER BY 句やインデックスの指定は できない。Twitter API には、一定時 間あたりの通信回数に上限が存在するか らである。一定回数を超えて通信を行う と、【図 22】のようなエラーが表示され、 しばらく接続不能になる。

たとえば ORDER BY 句を SQL で指 定すること自体は可能だが、その SQL を実行すると、世界中からリアルタイム で流れ込む大量のツイートを取得・選別 し続け、結果を表示するより先に、数秒 〜数十秒のうちに通信回数の上限に到達 してエラーになる。

データを並べ替えたい場合は、 TStringListやTClientDataSetなどの FireDAC接続から切り離されたローカ ルキャッシュ内に一度データを格納して から行うとよい。

なお、FireDACのデフォルト設定で はデータを1回の通信で50件ずつ取得 するため、キーワードに対する検索結果 が多い場合は、数分前までの新しいツ

🖾 Form1		
	* * * * * * * * * * * * * * * * * * * *	
	<b>E</b>	┃ Delphi/400℃IBM (に接続し、   デークな取得するついポークいた群
FDConnection1 FDQuery	DataSource 1	ナーダを取得するコンホーネント辞
FDPhysCDataCSVD	riverLink1	
		FDPhysCO400DriverLink1
Enterprise Connect	orsでCSVに接続し、	
データを読み書きす	るコンポーネント群	FDConnection2 FDQuery2
SELECT * FROM WORKC	SV	① SQL SELECT
	の受注データをWORKCSVに	② 登録
	の受注データをWORKCSVI() パラメーク設定)	
顧客□-ド=: CSV操作PGM②(接続/	の受注データをWORKCSVI。 ペラメータ設定) nection11	
顧客□ード=: CSV操作PGM②(接続) © FireDAC 接続エディタ - [FDCor ドライバまたはオーバーライドする接 ドライバまたはオーバーライドする接 ドライバまたはオーバーライドする接	の受注データをWORKCSVI( <b>ペラメータ設定</b> ) nection1] 振定義の名前を選択してから、パラメータをセ	②登録 
顧客コード=: CSV操作PGM②(接続) ◎ FireDAC 接続エディタ - [FDCor ◎ FireDAC 接続エディタ - [FDCor ◎ FireDAC 接続エディク - [FDCor ○ FireDAC 接続エディク - [FDCor     ]     ○ FireDAC #	の受注データをWORKCSV(こ <b>ペラメータ設定)</b> nection1] 航定義の名前を選択してから、パラメータをセ	②登録 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
<ul> <li>         ・ 顧客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVI。 ペラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセ 7ト	②登録 ットアップします 「CDataCSV」を選択すると 冬パラメータの一覧が読み込まれる
<ul> <li>         ・ 顧客コード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVI( パラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセ 7ト	②登録 ットアップします 「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSV(こ パラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセ 7ト] v ~	◎登録 ットアップします 「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる
研客コード=:   CSV操作PGM②(接続)   Simple Solution   FireDAC 接続エディタ - [FDCor    FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor    FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor    FireDAC 接続エディタ - [FDCor   FireDAC 接続エディタ - [FDCor	の受注データをWORKCSVI。 <b>ペラメータ設定)</b> nection1] 転定義の名前を選択してから、パラメータをセ クト v ・ 、 デフォルトに戻す(R) へルブ(H) 値	②登録 ットアップします 「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる 「 デフォルト
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSV(こ パラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセ クト v ~ (前 100	②登録 >>F79プします 「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる デフォルト
<ul> <li>         研客コード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVIL ペラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセクト ッ ッ ッ パーレングロ (値 100 デフォルトに戻す(図) ヘルングロ 「値 100 デフォルトに戻す(図) ヘルングロ 「値 100 「デフォルトに戻す(図) ヘルングロ 「個 100 「デフォルトに戻す(図) 「クリーングロ 「日まを	②登録 ットアップします 「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる 「 アフォルト」 100 100 100 100 100 100 100 10
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSV(こ パラメータ設定) nection1] 航定義の名前を選択してから、パラメータをセ 7ト] v へルブ(1) 値 100 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx	②登録         ットアップします         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         第発端末以外の端末で動作させる場合は、 「deployment licensing tyt !!=記載された
<ul> <li>         研客コード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVIと ペラメータ設定) nection1] 転産義の名前を選択してから、パラメータをセ クト マ アフォルトに戻す(2) イルブ(4) 値 100 次000000000000000000000000000000	②登録         ットアップします         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         「デフォルト」         「アクカルト」         「別発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された。         ランタイムキーを「RTK」に指定する必要がある
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVI ペラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセ クト ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	②登録         >hTPy7Ust         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         アフォルト         第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された。 ランタイムキーを「RTK」に指定する必要がある         Fake
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVL         パラメータ設定)         nection1]         定義の名前を選択してから、パラメータをセクト         ア         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ	②登録         ットアップします         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         アフォルト         「アフォルト         「アフォータの一覧が読み込まれる         「deployment_licensing.txt」に記載された。         ランタイムキーを「RTK」に指定する必要がある         「品を         書き込みを有効にする場合は、
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVIL パラメータ設定) nection1] 転定義の名前を選択してから、パラメータをセクト アレージェンクト アレージョンクト アレージョンクト アレージョンクリージェンクト アレージョンクト アレージョンクリージョンクト アレージョンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクリンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクト アレージョンクリンクリンクリンクリンクリンクト アレージョンクリンクリンクリンクリンクリンクリンクリンクリンクリンクト アレージョンクリンクリンクリンクリンクリンクリンクリンクリンクリンクリンクリンクリンクリン	<ul> <li>② 登録</li> <li>&gt;&gt;h7&gt;j/L\$\$</li> <li>CDataCSVJを選択すると 各パラメータの一覧が読み込まれる</li> <li>第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された ランタイムキーを「RTK」に指定する必要がある</li> <li>書き込みを有効にする場合は、 「UseRowNumbers」をTruelに指定</li> </ul>
<ul> <li>         研客□ード=:         <ul> <li></li></ul></li></ul>	の受注データをWORKCSVIL       パラメータ設定)       nection1]       航空義の名前を選択してから、パラメータをセクト       クレーク       クレーク <td>②登録         &gt;hF7y7Lます         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         第名パラメータの一覧が読み込まれる         第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された。 ランタイムキーを「RTK」に指定する必要がある         Fase         書き込みを有効にする場合は、 「UseRowNumbers」をTruelこ指定</td>	②登録         >hF7y7Lます         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         第名パラメータの一覧が読み込まれる         第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された。 ランタイムキーを「RTK」に指定する必要がある         Fase         書き込みを有効にする場合は、 「UseRowNumbers」をTruelこ指定
田客コード=:     田客コード=:     日本のので、「「「「「」」」」」」、     田本のので、     田本ののので、     田本のので、     田本ののので、     田本のので、     日本のので、     日本のので、     日本のので、     日本のので、     日本ののので、     日本のので、     日本のので、     日本のので、     日本のので、     日本のので、     日本のので、     日本ののので、     日本のののでの     日本ののので、     日本ののので、     日本ののので、     日本ののので、     日本ののので     日本ののので     日本ののので     日本ののので     日本ののので     日本ののので     日本ののので     日本ののので     日本ののののののののののののの     日本のののの     日	の受注データをWORKCSVIL         ペラメータ設定)         nection1]         転主義の名前を選択してから、パラメータをセクト         クト         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         グ         「信         100         XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	② 登録         ットアップします         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         アフォルト         アフォルト         第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された ランタイムキーを「RTK」に指定する必要がある         Fase         書き込みを有効にする場合は、 「UseRowNumbers」をTruelこ指定
田客コード=:     田客コード=:     日本ののでは、「「「「「「」」」」」」、     日本ののでは、     日本のので	の受注データをWORKCSVL         パラメータ設定)         nection1]         航空義の名前を選択してから、パラメータをセクト         クト         グ         グト         グ         グ         グ         グン         「信         100         X000000000000000000000000000000000000	② 登録         ットアップします         「CDataCSV」を選択すると 各パラメータの一覧が読み込まれる         アフォルト         第発端末以外の端末で動作させる場合は、 「deployment_licensing.txt」に記載された。 ランタイムキーを「RTK」に指定する必要がある         『書き込みを有効にする場合は、 「UseRowNumbers」をTruelに指定         2019#PGMWC12#

イートしか表示されないこともある。 このような場合には、【ソース 5】で 指定したように、TFDQueryの FetchOptions.RowsetSizeの設定値を 増やすことで、1回の通信あたりの取得 レコード数が増え、より多くの取得結果 が得られる。

ここまで Twitter API を利用した データ送信およびデータ参照の手順を紹 介したが、CData Twitter FireDAC Components では、目的に合わせたデー タを解析できるよう、さまざまなテーブ ルやビューが準備されている。代表的な ものを、【図 23】に記す。

#### 5-2. Googleスプレッドシートへの データ出力

次に、「CData Google Sheets FireDAC Components」を使用して、IBM i の売 上データを「Google スプレッドシート」 に出力する例を紹介する。また、出力し た Google ドライブ上のスプレッドシー トを、ローカルファイルとしてダウン ロードする方法もあわせて紹介する。

これまで IBM i のデータを出力する 場合、ローカル PC 上に CSV ファイル や Excel ファイルで出力することが多 かったが、クラウド上の Google スプレッ ドシートにデータが出力できれば、ファ イルの共有や同時編集など利便性が向上 する。

Delphi/400 10.2 Tokyo を起動した ら、アプリケーションを新規作成し、新 しいフォームに、【図 24】のようにコン ポーネントを配置する。

前章と同様、Delphi/400 で IBM i に 接続してデータを取得するコンポーネント 群と、Enterprise Connectors で Google スプレッドシートに接続するコンポーネ ント群をそれぞれ配置する。

なお Google スプレッドシートで使用 するコネクターは、「TFDPhysCDataG oogleSheetsDriverLink」である。

前章と同じように、TFDConnection をダブルクリックして FireDAC 接続エ ディタを開き、ドライバID に 「CDataGoogleSheets」を選択すると、 【図 25】のように接続時の各種パラメー タが自動でセットされる。本稿では、現 在使用中の PC でログインしている自身 の Google アカウントでの接続を前提と するため、【図 25】に記載の設定のみ行 えばよい。

FireDAC接続エディタの画面で、設 定変更後に「テスト」ボタンを押すと、 ブラウザが起動して【図 26】のような 認証画面が表示されるので、このタイミ ングで Google アカウントへの権限を許 可しておく(認証が必要なのは初回接続 時のみ)。

この認証により、各処理はログイン中 の自身の Google アカウントとして行わ れる。したがって、完成したアプリケー ションを別の Google アカウントでログ インした別の PC から起動した場合、初 回接続時に同じようにブラウザが立ち上 がり、【図 26】の認証画面が表示される。 本稿のサンプルでは、前章で CSV に

データを出力した際と同じデータを使用 して、【ソース 6】のようにロジックを 記述する。

スプレッドシートの新規作成では、 SQL ではなくストアドプロシージャを 発行している。Enterprise Connectors では、データの読み書きだけではなく、 ファイルの作成や削除、ダウンロードな どの処理が行えるようサブメソッドがコ ネクターごとに用意されており、使用で きる。

TFDQueryでもストアドプロシー ジャの発行は可能だが、【ソース6】の ようにTFDStoredProcで発行するのが シンプルで効率がよい。Googleスプレッ ドシートの操作には、今回使用したスプ レッドシートの新規作成以外にも、シー トの追加・コピー・削除や書式設定など、 ストアドプロシージャで実現できるメ ソッドが数多く存在するので、ぜひヘル プドキュメントで詳細を参照いただきた い。

また本稿のプログラムでは、出力先ス プレッドシートの「A」~「F」という 列番号がそのままフィールド ID となっ ており、SQL で INSERT 文を発行する 際にもそのように記述する。

完成したプログラムを実行して、「① 出力」ボタンを押すと、ログイン中の Google アカウントが保持する Google ス プレッドシートの一覧にファイルが新規 作成され、その中に IBM iから取得した データがセットされる。【図 27】【図 28】 また、【ソース7】のようにダウンロー ドボタンのロジックを記述することで、 このスプレッドシートをプログラム内か ら直接 (ブラウザを起動せずに)、xlsx や PDF といった形式でダウンロードで きる。【図 29】【図 30】

ここでも、ダウンロードの実行にはス トアドプロシージャを使用している。

#### 5-3. Google Driveでのデータ検索

最後に、「CData Google Drive FireDAC Components」を使用した、 Google Driveのファイル検索について 紹介する。

Delphi/400 10.2 Tokyo を起動した ら、アプリケーションを新規作成し、新 しいフォームに【図 31】のようにコン ポーネントを配置する。

このプログラムでは IBM i と接続し ないため、Enterprise Connectors で Google Drive に接続するコンポーネン ト群のみをそれぞれ配置する。なお Google Drive で使用するコネクターは、 「TFDPhysCDataGoogleDriveDriverLi nk」である。

前 章 ま で と 同 じ よ う に、 TFDConnection をダブルクリックして FireDAC 接続エディタを開き、ドライ バ ID に「CDataGoogleDrive」を選択 すると、【図 32】のように接続時の各種 パラメータが自動でセットされる。

テスト接続時では、前章の Google ス プレッドシートと同様に、【図 26】のよ うな Google アカウントの認証画面が初 回のみ表示されるので、権限を許可する。

次に、【ソース 8】のようにロジック を記述する。

Google Drive の検索機能では、通常 のファイル名やファイルの作成日時など による絞り込みだけでなく、ファイルの 内容を含むフルテキスト検索にも対応し ている。

たとえば前項で作成した Google スプ レッドシートで、顧客名に「株式会社吉 田商事」という値をセットしたセルが存 在するが、Google Drive 側で検索条件 に「吉田商事」と入力することで、セル の内容に「吉田商事」が含まれるスプレッ ドシートを検索結果として表示させられ る。【図 33】

これは Google スプレッドシートの ファイル保存先が Google Drive になっ

ノアイル名はて		<ul> <li>WORKCSV.csv メモ糖</li> <li>ファイル(F) 編集(E) 雷</li> </ul>	。  式(0) 表示(V) ヘルプ(H	)	-C-)	
		"YYTRC1","Y#TF02"," "'0301","株式会社言 "0308","太平洋食品	YYTRC3″,″WLTRO4″,″\ 征田商事″,″3フインチ  性式会に″,″ヘッドオ	/WTRO5 <sup>**</sup> , <sup>**</sup> /WTRO0 <sup>**</sup> ·液晶モニター <sup>**</sup> , <sup>**</sup> , <sup>**</sup> 2 <sup>**</sup> , ·ンステレオ <sup>**</sup> , <sup>**</sup> , <sup>**</sup> 73	"48000","98000" 380","7980"	<u>*</u>
2行目以降が	各レコードの	~ 0002,小你带偏梁 ~ 0007","埼玉茂事挑 ~ 0001","株式会社言 ~ 0008","太平洋食品	«休式会社", "インスリ (式会社", "ヘッドホン (田蘭事″, "新イクラ丼  株式会社", "フーブレ	ンデコーン,11,1987 ステレオブ,11,17,7987 キャッ~″,110″,729807 用感熟紙″,1207,7983	JC00 , 2000  ″,″7960″ /"″298C0″ )″,″ 960C″	E
			†手 産業", 13 色ボール 「業デンキ", 13 フィン 記岐電様", 7町 イクラ丼	ベン","100~,"240″, (手液晶で出身・","1 (キット","5″,"2380″, (10ペン", "2480″,	124000" ","40000","40000" ,"14900" "	-
		(************************************	田商事","全自動洗濯 販売株式会社","ワー 株式会社","電気ジャ	· 秋", *1″, *95800*, *91 · ブコ用感熱紙″, *10″, - ポット*, *2″, *148	9800~ ,~280~,~2800~ )0~,~28600~	
		0007, 每玉做手條 [10]ULE,"許馬事務機 [10]041,"株式会社岩	(現去社 , ジルーラッ (株式会社**, **全日動洗  手産業**, **インボート	(と時1,1 ,98 ,98 深理機1,131,1998JU1, 下シャツ1,101,193	"259400" 20°, "15000"	-
-71						
FireDAC拐	き続したCSVから	SELECTでデ	ータ抽出			
{************************************	************************************	****	*****			
procedure TForm1. var	Button2Click(Sender: TObjec	t) :	*****			
i: Integer; Begin						
FDQuery1.Close: FDQuery1.SQL.Te	xt := Edit1.Text; // SELE	CT * FROM WORKCSV		┃ この3行だけ <sup>・</sup> ┃ SQLを発行し	でSELECTの 、DBと同じよう(	=
T bouer yr. open.				- CSVの内容を	を影できる	
// 列幅の調整						
// 列幅の調整 for i := 0 to ( begin DBGrid1 Colum	DBGrid1.Columns.Count - 1)	do				
// 列幅の調整 for i := 0 to ( begin DBGrid1.Colum end; end;	DBGrid1.Columns.Count - 1) ns.Items[i].Width := 80;	do				
// 列幅の調整 for i := 0 to ( begin DBGrid1.Colum end; end;	DBGrid1.Columns.Count - 1) ns.Items[i].Width := 80;	do				
// 列幅の調整 for i := 0 to ( begin DBGrid1.Colum end: end:	(DBGrid1.Columns.Count - 1) ns.Items[i].Width := 80;	do				
// 列幅の調整 for i := 0 to ( begin DBGrid1.Colum end; end; 8 CSV操作PGI	(DBGrid1.Columns.Count - 1) ns.Items[i].Width := 80; M④(SELECTで内容表	do 毛示)				
<ul> <li>// 列幅の調整 for i := 0 to 0 begin DBGrid1.Colum end: end:</li> <li>8 CSV操作PGI</li> <li>1 Form1</li> </ul>	(DBGrid1.Columns.Count - 1) ns.Items[i].Width := 80; M④(SELECTで内容表	do ₹示)				
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; 8 CSV操作PGI	(DBGrid1. Columns. Count - 1) ins. Items[i]. Width := 80;	do 表示) TR04 WWTR05 W 2 48000 1 7980	WTR06 RowNumbe 96000 7980			
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; end: 8 CSV操作PGI	(DBGr id1. Columns. Count - 1) Ins. Items [i]. Width := 80;	do 毛示) TR04 WWTR05 W 2 48000 1 7980 3000 7980	WTR06 RowNumbe 96000 7980 3000 7980			
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10001 10002 10007 CSVの1行	(DBGr id1. Columns. Count - 1) Ins. Items [i]. Width := 80; M④ (SELECTで内容表 WWTR02 WWTR03 WWT 株式会社吉田和 37インチ液晶モ 太平洋食品株式 ヘッドホンステレス 小林常備業株式 インスタントコーと 埼玉商事株式 ヘッドホンステレス 1日の値がそのままフィー 日、10路 が そのままフィー	do 表示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 ールドIDになる いたちて	WTR06 RowNumbe 96000 7980 3000 7980 29800 19600 19600 19600		として、内部	
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10007 CSVの1行 CSVの2行 10005	(DBGr id1. Columns. Count - 1) Ins. Items [i]. Width := 80:	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIになる ードとなる 5 2980	WTR06 RowNumbe 96000 7980 3000 7980 29800 19600 24000 24000 24000 24000 14900	アリング 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	として、内部 berJができる	
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10005 10005 10005 10002 10001	(DBGr id1. Columns. Count - 1) Ins. Items [i]. Width := 80;	do ま示) TR04 WWTR05 W 2 48000 1 7980 1 7980 3000 7980 ールドIDIになる ードとなる 5 2980 200 240 1 99900	WTR05 RowNumbe 96000 7980 3000 7980 29800 19600 24000 14900 14900 14900 48000 99900	アリング 10 11	として、内部 berJができる	
// 列幅の調整 for i := 0 to 0 begin DBGridl.Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10003 10007 CSVの1行 CSVの2行 10005 10002 10001 10003 10003	(DBGr id1. Columns. Count - 1)         ins. Items [i]. Width := 80;         M④(SELECTで内容表         WWTR02       WWTR03         WWTR02       WWTR03         水平洋食品株5       ヘッドホンステレマ         小林常備薬株5       ソスタントコーと         埼玉商事株式5       ヘッドホンステレマ         日の値がそのままフィー       日以降が行単位でレコー         株式会社茨城1       新イクラ井キット         小林常備薬株5       3色ポールペン         株式会社茨城1       新イクラ井キット         小林常備薬株5       3色ポールペン         株式会社茨城1       新イクラ井キット	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIこなる ードとなる 5 2980 200 240 1 99800 1 9800 1 98000 1 9800 1 9800	WTR05 RowNumbe 96000 7980 29800 7980 29800 19600 24000 14900 14900 14900 14900 48000 99900 9800 9800 29600	アリング 11 12 13 14 14 14 14 14 14 14 14 14 14 14 14 14	として、内部 ber」ができる	
// 列幅の調整 for i := 0 to 0 begin DBGrid1.Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10001 10002 10007 10005 10002 10001 10003 10003 10003 10007 10006 10006	(DBGr id1. Columns. Count - 1)         Ins. Items [i]. Width := 80;         (4) (SELECTで内容表         (5) (SELECTで内容表         (4) (SELECTで内容表         (5) (SELECTで内容表         (7) (SELECTで内容表	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIこなる -ドとなる 5 2980 200 240 1 99900 10 980 2 14800 1 9880 2 14800 1 9880 2 14800	WTR05 RowNumbe 96000 7980 3000 7980 19600 19600 24000 148000 148000 148000 148000 99800 98800 29600 9880 299400	ア 用ユニークキー -ルド「RowNumb 9 10 11 12 13 14 15	として、内部 berJができる	
// 列幅の調整 for i := 0 to 0 begin DBGrid1. Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10003 10007 CSVの1行 CSVの2行 10005 10002 10003 10003 10003 10003 10004 10004 10004 10004	(DBGr id1. Columns. Count - 1)         Ins. Items [i]. Width := 80;         (4) (SELECTで内容表         (5) (SELECTで内容表         (4) (SELECTで内容表         (5) (SELECTで内容表         (7) (SELECTでいっっっっっ)         (7) (SELECTでのっっっ)         (7) (SELECTでいっっっっっっ)         (7) (SELECTC)         (7) (SELECT)         (7) (SELECT)	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIこなる -ドとなる 5 2980 3000 7980 1 99800 1 9980 1 9980 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 9880 1 98	WTR05 RowNumbe 96000 7980 3000 29800 19600 24000 14900 48000 14900 48000 99900 9800 29600 9800 29600 980 299400 19800 3000	ア 用ユニークキー -ルド「RowNumb 9 10 11 12 13 14 15 16 17 、	として、内部 ber」ができる	
// 列幅の調整 for i := 0 to 0 begin DBGrid1.Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10002 10007 10005 10002 10007 10001 10003 10002 10001 10003 10002 10001 10003 10002 10001 10003 10003 10003 10004 10004 10004 10005 10005 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1006 1007 1007 1007 1006 1007 1007 1006 1007 1007 1008 1007 1007 1008 <td>(DBGr id1. Columns. Count - 1)         ins. Items [i]. Width := 80;         (4) (SELECTで内容表         (5) (SELECTで内容表         (7) (SELECTでいっ         (7) (SELECTで内容表         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTC)         (7) (SELECTC)</td> <td>do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIこなる -ドとなる 5 2980 200 240 1 99800 10 980 2 14800 1 99800 10 980 2 14800 1 99800 1 99800 1 99800 1 99800 1 99800 1 3000</td> <td>WTR05 RowNumbe 96000 7980 3000 29800 19600 24000 19600 24000 148000 48000 99800 14900 48000 99800 98800 29600 9880 299400 19800 3000</td> <td>ア ア</td> <td>として、内部 ber」ができる</td> <td></td>	(DBGr id1. Columns. Count - 1)         ins. Items [i]. Width := 80;         (4) (SELECTで内容表         (5) (SELECTで内容表         (7) (SELECTでいっ         (7) (SELECTで内容表         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTでいっ         (7) (SELECTC)         (7) (SELECTC)	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIこなる -ドとなる 5 2980 200 240 1 99800 10 980 2 14800 1 99800 10 980 2 14800 1 99800 1 99800 1 99800 1 99800 1 99800 1 3000	WTR05 RowNumbe 96000 7980 3000 29800 19600 24000 19600 24000 148000 48000 99800 14900 48000 99800 98800 29600 9880 299400 19800 3000	ア ア	として、内部 ber」ができる	
// 列幅の調整 for i := 0 to 0 begin DBGrid1.Colum end; end; 8 CSV操作PGI 8 CSV操作PGI 10001 10001 10003 10002 10005 10005 10005 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10006 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10007 10008 10009 10008 10009 10009 10008 10008 10009 10009 10008 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009 10009	(DBGr id1. Columns. Count - 1)         ins. Items [i]. Width := 80;         M(④)(SELECTで内容表         WWTR02       WWTR03         WWTR02       WWTR03         WWTR02       WWTR03         WHTR02       WWTR03         WHTR02       WWTR03         WHTR03       WWTR04         WHTR04       ST/27/26/26/27/26/26/26/26/26/26/26/26/26/26/26/26/26/	do ま示) TR04 WWTR05 W 2 48000 1 7980 3000 7980 -ルドIDIになる -ドとなる 5 2980 200 240 1 99900 1 99800 1 99900 1 99800 1 99900 1 99800 1 9	WTR06 RowNumbe 96000 7980 3000 7980 19600 29800 19600 24000 19600 24000 14900 48000 14900 48000 99800 9800 9800 9800 9800 980	ア 用ユニークキー ア ルド「RowNumb 9 10 11 12 13 14 15 16 17 ・	として、内部 berJができる	

ているためで、同じ Google アカウント であれば、Google Drive 上からもスプ レッドシートの内容を参照できる。

前章で、スプレッドシートを新規作成 した際に採番されたファイルの ID も Google Drive と共通なので、その ID を含んだ Google Drive のファイル URL を指定して実行することで、Google ス プレッドシートで対象のファイルが起動 する。【ソース 9】【図 34】

Google Drive ではこれ以外にも、各 種 SQL やストアドプロシージャを使用 することで、ファイルのアップロード、 ダウンロードやファイル名の変更・削除 なども行える。これらの方法もそれぞれ ヘルプドキュメントに記載されているの で、ぜひ一度挑戦していただきたい。

## 6. まとめ

本稿では、Enterprise Connectors を 使用したクラウドサービスとの連携テク ニックについて紹介した。

今回は Twitter、Google スプレッド シート、Google Drive という一般によ く使用されるクラウドサービスを題材に 連携方法を紹介したが、Enterprise Connectors では他にも Salesforce や kintone といったエンタープライズ向け クラウドサービス や、MongoDB、 Amazon DynamoDB といったような NoSQL データベースなど、これからの 業務システムに多用する多彩なサービス へのアクセスを可能にしている。

本稿を参考に、さまざまなクラウド サービスとの連携を検討いただきたい。 なお、全コネクターが利用可能な Enterprise Connectors Plusの購入を ご検討の場合は、弊社営業までお気軽に お問い合わせいただければ幸いである。

Μ



তাত	CSV操作PGM⑤(CSVへ	ØINSERT)
5		
Γ	顧客コード= 10004 の受注データ?	EWORKCSVIZ ② 登録
ſ	WORKCSV.csv XTHE	
	ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ( "10:17"、"埼玉商事件八会社","新イクラ3 "10:15"、"株式会社"の時期 " ベッド市	H)
	- 101111、"株式会社告証商事"、"ウルーシロ 11013"、"株式会社手業デンキ"、『インス - DU-22"、「小林軍旗殺婆式会社」、「21-1	が全義が、"Son"、"Autor"、 タンドコービーで、"I("、 「、"30000" 「月感教題"、"Tu"、"(SU U)」、
	- 110-14、 本京史務後期に株式会社 , デジ 11011351、『東京事務後期庁株式会社", デジ 「1101135」、「詳馬事務後承式会社"、「インボ、 1101171、『愛玉商事株式会社1、『フルーン(	/1.//7.#2年1971 (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997) (1997)
	「1023」、「辞馬事務法体式会社」、「なーン」 「1021」、「株式会社吉田蘭事」、「商品的」、 「1011」、「作業学会社主日高事」、「「な」」、「1 「1024」、「株式会社去日音書」、「3、11」」	リン紅茶(), ***, **200 3**, *) *** ****************************
	110日41、"林式会社告手産業", "インボー 110日41、"林式会社告手産業", "インボー "10日41、"林式会社告手産業", "インボ	トージャップ(1し,,,,,,,,,,,,,,,,,,)) (1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
L		
ſ	登録結果②	10001 株式会に書去には感名 3 6 0 00 10001 株式会社 去王石 5 6 0 37 10001 株式会社 去王石 5 6 0 37 10001 株式会社 去王石 7 240 2400 38 5 10001 株式会社 会社 10001 10 10 10 10 10 10 10 10 10 10 10
L	明神を開き直りと、打か追加	211 ( いる) 10 ( 10 ( 10 ( 10 ( 10 ( 10 ( 10 ( 10
図10	) Twitter Web APIの利用	開始①
	🔰 Developer Use cases	Products Docs More Labs Apply Apps 📆
	Аррз	Create an app
	Apps	Create an app
	Apps No apps here. You'll need an app and API key in orde	r to authenticate and integrate with most Twitter developer products. Create
	Apps No apps here. You'll need an app and API key in orde	Create and integrate with most Twitter developer products. Create 「Create an app」をクリックし、 今回作成するアプリケーション用のキーを取得する
	Apps No apps here. You'll need an app and API key in orde	Create and integrate with most Twitter developer products. Create 「Create an app」をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)
	Apps No apps here. You'll need an app and API key in orde	Create an app r to authenticate and integrate with most Twitter developer products. Create 「Create an app」をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)
図11	Apps No apps here. You'll need an app and API key in orde Twitter Web APIの利用	Create and integrate with most Twitter developer products. Create r to authenticate and integrate with most Twitter developer products. Create 「Create an app」をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提) 開始②
図11	Apps No apps here. You'll need an app and API key in orde Twitter Web APIの利用	Image: Create and integrate with most Twitter developer products. Create
図11	Apps No apps here. You'll need an app and API key in orde Twitter Web APIの利用 Y Developer Use cases Pro	Image: Construction of the same ways through the same.
図11	Apps No apps here. You'll need an app and API key in orde Twitter Web APIの利用 Developer Use cases Pro welcome We have	Image: Create and integrate with most Twitter developer products. Create       Image: Create an app.]をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         開始②         duers       Deex More       Labs       Deexboard       Image: Create apps twitter.com. You can manage any of your existing apps in all of the same ways through the safe.
図11	Apps No apps here. You'll need an app and API key in order Twitter Web APIの利用 Developer Use cases Pro 愛 #welcome We have Junderstanding apps	Image: Create and integrate with most Twitter developer products. Create       「I key.         Image: Create an app]をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         開始②         Cuers Des Mere Labs       Desnboard         e sunset apps twitter.com. You can manage any of your existing apps in all of the serie ways through this site.       Image: Create app details         The following app details will be visible to app use (アプリケーション名)       アプリケーション名
図11	Apps No apps here. You'll need an app and API key in order Twitter Web APIの利用 Developer Use cases Pro ジ Developer Use cases Pro ジ Welcome We have What is an app? ヘ	Create an app       Create an app         Create an app       をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         開始②         curs       Des         Mer       Labs         Databoard       Imp         e sunset apps byttler com. You can manage any of your existing apps in all of the same ways through this site.       Imp         App details       Imp         Brotowing up details will be visible to app use Arroll phot—Spin 2000       Imp         Imp of the API happen mediate to a pp use Arroll phot—Spin 2000       Imp         Imp of the API happen mediate to a underrice to the API h
図11	Apps No apps here. You'll need an app and API key in order Twitter Web APIの利用 Protection Use cases Pro 愛 Rwelcome We have What is an app? ^ Think of a Twitter app as a gateway. It provides a set of authorized and the set of authorized and the set of	Create an app」をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         開始②         cutts       Dec         Mer       Lab         Papetalls         Productalls         Productall         Productall         Productall         Productalls         Prod
図11	Apps No apps here. You'll need an app and API key in order Twitter Web APIの利用 Protection Web app の Pro Welcome We have What is an app? ^ Think of a Twitter app as a gateway. It provides a set of authentication keys and permission settings needed for making requests to most Twitter	Create an app.jをクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         Impace         Create app.s byther ones to use one one one one one one one one one on
図11	Apps No apps here. You'll need an app and API key in order Twitter Web APIの利用 Protections View Developer View Developer Protections Protections View Developer View	Create an app       Create an app         Coreate an app」をクリックし、 今回作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)         開始②         Coreate an app app app app app app app app app
図11	Apps No apps here. You'll need an app and API key in order  Twitter Web APIの利用  Vectors Pro Vectors P	Constant and integrate with most Twitter developer products. Creat       「ky,         Constant and ppl Schlynych, 今日作成するアプリケーション用のキーを取得する (※本稿では、Twitter開発者アカウントは取得済みの前提)
図11	Apps No apps here. You'll need an app and API key in order  Twitter Web APIの利用  Proveloper Use cases Pro  Pro  Pro  Pro  Pro  Pro  Pro  Pro	Create an app         Create app
図11	Apps No apps here. You'll need an app and API key in order  Twitter Web APIの利用  Pereloper Use cases Pro  Pro  Pro  Pro  Pro  Pro  Pro  Pro	Image: Contract and integrate with most twitter developer products. Creat       「ky,         Image: Contract and integrate with most twitter developer products. Creat       「ky,         Image: Contract and projection of the state of the sta
図11	Apps No apps here. You'll need an app and API key in order  Twitter Web APIの利用  Pereloper Use cases Pro  Pro  Pro  Pro  Pro  Pro  Pro  Pro	Create an apply をかりやかし、 の当れたするアブリケーション用のキーを取得する (沙木箱では、Twitter開発者アカウントは取得済みの前提)    開始②          active are an apply をかりやかし、 の当たすでは、Twitter開発者アカウントは取得済みの前提)          phata@    (active are an apply and the state are and and are are and a dot the same ways through that suff.          phata@    (b) Diversion and and a dot the same ways through that suff.          phata@    (b) Diversion and and a dot the same ways through that suff.          phata@    (c) Diversion and and a dot the same ways through that suff.          phata@    (c) Diversion and and a dot the same ways through that suff.          phata@    (c) Diversion and and a dot the same ways through that suff.          phata@    (c) Diversion and the subside to use set this is a dot the same set this is a dot dot the same set this a dot the same set the
図11	Apps No apps here. You'll need an app and API key in order  Twitter Web APIの利用  Developer Use cases Pro  Comparison Settings needed for making requests to most Twitter APIs.  Why register an app?  To use an API, we require you create an app as part of our DAuth authorization schema.  Which products require an API ~	Create an apply Editive average products Creat       Likey.         Create an apply Editive developer products Creat       Likey.         Create an apply Editive developer products Creat       Likey.         Create an apply Editive developer products Creat       Likey.         Breat       Developer developer developer products Creat         Breat       Developer devel

\_\_\_\_

\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

図12 Twitter Web APIの利用開始③	
💆 Developer Use cases Products Does More Lats. Deshbeard	
😸 Twelcome We have some tapped with examply in can manage any of your existing apps in all of the same ways through this site. 🗙	
Apps > MBGARO TEST App details Keys and tokens Permissions	
App details     Detail and URLs	
App Norm E	
SAMPLE FOR MGR TECH REP 12 Description	
Website URL	
https://www.migaro.co.jp/ Sign in with Twitter	
Uisabled T	
図13 Twitter Web APIの利用開始④	
👽 Developer Use cases Moducis Ducs Mune Labs Dashiboand 🗊	
Avalcome We have subset apps trutter com. You can manage any of your existing apps in all of the same ways through this ana. X	
Apps > MIGARO TEST App details Keys and tokens Permissions	
Keys and tokens Generate (またはRegenerate)ボタンで、	
Koya. socret koya and access tokkons management. Consumer API keys	
5 (AT key)	
Access token & access token secret	
et u (Access Joner) F v (Access Joner) Rend and write (Access Isken sourch)	
(Regenciale)	
図14 Twitterのデータ送信①(画面設計)	
© Form1	
Enterprise ConnectorsでTwitterに接続し、	
デーダを読み書きするコンホーネンド群	
FDConnection1 FDQuery1	
FDStoredProc1	
FDPhysCDataTwitterDriverLink1 FDConnection2 FDQuery2	
① 商品マスタ検索 ② 選択行の商品についてツイートする ② ツイート結果をブラウザで確認する	
Delphi/400でIBM i に接続し、 データを取得するコンポーネント群	

■ ドライバまたはオーバーライドする				
	5接続定義の名前を選択してから、パラメータを	セットアップします		
	レクリプト	[ODetaTut	はないた。蜜ヤナフリ	L-
	taTwitter 🗲	というメー	uer」で迭灯9つ0 タの一覧が詰み、	
				20100
接続定義名(1):		<b>T</b>		
テストロ ウィザー	<u>*(W) 「デフォルトに戻す(R)」 へルプ(</u>	<u>H</u>		
パラメータ	値		デフォルト	*
DriverID	CDataTwitter		CDataTwitter	
CallbackURL				
InitiateOAuth	OFF		OFF	
OAuthAccessToken				
OAuthAccessTokenSecret				
OAuthClientId				
OAuthClientSecret	BY ADDOATA BY WYCD-L-Y		8/ 4000 4 TA // 1000- 1-100	Suitten Date General 1923 4
Account	%APPDATA%##CData#	**   Witter Data Provider##0A	76APPLIA I A 7644LUB13441	Witter Data Provider##UA
LizzAppOplyAuthoritication	Ealea		Cales	
AutoCache	False		False	
CarboCarpantian	1 disc		1 0/50	
Twitterのデータ送信(	③(接続パラメータ設定)			
Twitterのデータ送信( 目的別 TFDConnectionのパ	3)(接続パラメータ設定) ラメータ設定値>	7 / 10184 + 4= 2	ж [	」はデフォルト値のる
Twitterのデータ送信 目的別 TFDConnectionのパ 接続目的	3)(接続パラメータ設定) ラメータ設定値> 開発者自身のアカウントでのみ	ふみ/削除を行う	※「 トでログインして	」はデフォルト値の3 照会のみ
Twitterのデータ送信( 目的別 TFDConnectionのパー 接続目的 パラメータ名	3) (接続パラメータ設定) ラメータ設定値> 開発者自身のアカウントでのみ 書き込み ご制除を行う	ふみ/削除を行う あらゆるアカウン まき込み/	※「 ・ ドでログインして 割除を行う	」はデフォルト値の 照会のみ (書き込み/削除不要
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名	3) (接続パラメータ設定) ラメータ設定値> 開発者自身のアカウントでのみ 書き込み/削除を行う	A A A A A P P 作 成 時 に 取 得 し 、 、 、 、 、 、 、 、 、 、 、 、 、	※「 トでログインして 削除を行う ⊱「API key」	」はデフォルト値の 照会のみ (書き込み/削除不要
Twitterのデータ送信 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret	3)(接続パラメータ設定) ラメータ設定値> 開発者自身のアカウントでのみ 書き込み/削除を行う	A A A A P P 作 成 時 に 取 得 し た 「 の し 7 日 た た う 、 の ら ゆ る ア カ ウ ン 、 、 、 、 、 、 、 、 、 、 、 、	※「 トでログインして 削除を行う た「API key」 API secret key」	」はデフォルト値の。 照会のみ (書き込み/削除不要
Twitterのデータ送信( 目的別 TFDConnectionのパー 接続目的 パラメータ名 OAuthClientId OAuthClientSecret	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>第発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動せ</li> </ul>	A A A A A P P 作 成 時 に 取 得 し た 「 で た 「 で か ら ゆ る ア カ ウ ン み / A と ひ み / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / / A / A / A / / A / A / A / / A / A / A / / A / A / A / / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A / A /	※「 <b>トでログインして</b> 削除を行う た「API key」 API secret key」 均場所に記載されたラ	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定
Twitterのデータ送信( 目的別 TFDConnectionのパー 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> </ul>	APP作成時に取得した「 たさせる場合は、以下の CData FireDAC Composite	※「 トでログインして 削除を行う た「API key」 API secret key」 つ場所に記載されたラ nents for Twitter¥der	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 ployment licensing txt
Twitterのデータ送信( 目的別 TFDConnectionのパラ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlvAuthentication	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>■書き込 開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>▲</li> <li>▲</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C     </li> </ul>	APP作成時に取得した「 たさせる場合は、以下の CData FireDAC Composite	※「 ドでログインして 削除を行う た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 ployment_licensing.txt True
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURI	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>■書き込 開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>AI</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> <li></li> <li></li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 作させる場合は、以下の CData FireDAC Compon APP作成時に指定す	※「 ・トでログインして 削除を行う た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」	」はデフォルト値の。 照会のみ (書き込み/削除不要 ンタイムキーを指定 oloyment_licensing.txt True 
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>         第発者自身のアカウントでのみ 書き込み/削除を行う     </li> <li>         A 開発端末以外の端末で動作 C:¥Program Files¥CData¥C     </li> <li> </li> </ul>	APP作成時に取得した などの場合は、以下の た APP作成時に取得した「 などする場合は、以下の CData FireDAC Compon APP作成時に指定す GETAND	※「 *トでログインして 削除を行う た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」 REFRESH	」はデフォルト値の。 照会のみ (書き込み/削除不要 ンタイムキーを指定 bloyment_licensing.txt True  
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>第発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動行 C:¥Program Files¥CData¥C</li> <li></li> <li></li></ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 たさせる場合は、以下の CData FireDAC Compon APP作成時に指定す GETAND	※「 <b>ドでログインして</b> 削除を行う と「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」 REFRESH	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 oloyment_licensing.txt True 
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>第発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token I</li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 存させる場合は、以下の Data FireDAC Compon APP作成時に指定す GETAND 接続時に都別	※「 トでログインして 削除を行う た「API key」 API secret key」 つ場所に記載されたラ nents for Twitter¥dep  る「Callback URL」 REFRESH 変関数で取得	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 oloyment_licensing.txt True   
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>■書き込 開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>▲</li> <li>▲</li> <li>第発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token」</li> <li>APP作成時に取得可能な</li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 存させる場合は、以下の Data FireDAC Compo CData FireDAC Compo APP作成時に指定す GETAND 接続時に都別 (詳細はヘルプド=	※「 <b>ドでログインして</b> <b>削除を行う</b> た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」 REFRESH 変関数で取得 チュメントを参昭)	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 oloyment_licensing.txt True   
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>■書き込</li> <li>開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>AA</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token」</li> <li>APP作成時に取得可能な 「Access token secret」</li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 なさせる場合は、以下の Data FireDAC Composi CData FireDAC Composi CData FireDAC Composi の APP作成時に指定す GETAND 接続時に都居 (詳細はヘルプドョ	※「 i i i i i i i i i i i i i	」はデフォルト値の。 照会のみ (書き込み/削除不要 ンタイムキーを指定 bloyment_licensing.txt True   
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken OAuthAccessTokenSecret	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>書き込</li> <li>開発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動行 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token」</li> <li>APP作成時に取得可能な 「Access token secret」</li> </ul>	A A A A P P 作 成 時 に 取 得 し た 「 本 さ せ る 場 合 は 、 以 下 の こ D ata Fire D A C D ata Fire D A C C の で の し た 「 本 さ せ る 場 合 は 、 以 下 の こ D ata Fire D A C C の p の に 取 得 し た 「 本 さ せ る 場 合 は 、 以 下 の こ つ - - - - - - - - - - - - -	※「 <b>*トでログインして</b> <b>削除を行う</b> た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」 REFRESH を関数で取得 Fュメントを参照) てAPIの認証画面が	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 bloyment_licensing.txt True    照会専用アカウントで
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken OAuthAccessTokenSecret 結果	<ul> <li>3)(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>第発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動行 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token」</li> <li>APP作成時に取得可能な 「Access token secret」</li> <li>開発者自身のアカウントで</li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 存させる場合は、以下の Data FireDAC Compon APP作成時に指定す GETAND 接続時に都居 (詳細はヘルプドョ プラウザが起動し 表示され、そご	※「 <b>トでログインして</b> <b>削除を行う</b> た「API key」 API secret key」 つ場所に記載されたラ ments for Twitter¥dep  る「Callback URL」 REFRESH を関数で取得 チュメントを参照) CAPIの認証画面が でログインした	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 bloyment_licensing.txt True   ローー 照会専用アカウントで ログイン(ユーザー西
Twitterのデータ送信( 目的別 TFDConnectionのパ 接続目的 パラメータ名 OAuthClientId OAuthClientSecret RTK UseAppOnlyAuthentication CallbackURL InitiateOAuth OAuthAccessToken OAuthAccessTokenSecret 結果	<ul> <li>③(接続パラメータ設定)</li> <li>ラメータ設定値&gt;</li> <li>第発者自身のアカウントでのみ 書き込み/削除を行う</li> <li>A</li> <li>開発端末以外の端末で動作 C:¥Program Files¥CData¥C</li> <li></li> <li></li> <li>APP作成時に取得可能な 「Access token」</li> <li>APP作成時に取得可能な 「Access token secret」</li> <li>開発者自身のアカウントで ログインし、読み書きする</li> </ul>	APP作成時に取得した「 ならゆるアカウン 書き込み/ APP作成時に取得した「 存させる場合は、以下の Data FireDAC Compon APP作成時に指定す GETAND 接続時に都服 (詳細はヘルプドョ プラウザが起動して 表示され、そこ アカウントで	※「 <b>ドでログインして</b> <b>削除を行う</b> た「API key」 API secret key」 つ場所に記載されたラ nents for Twitter¥dep  る「Callback URL」 REFRESH を関数で取得 キュメントを参照) CAPIの認証画面が でログインした 読み書きする	」はデフォルト値の 照会のみ (書き込み/削除不要 ンタイムキーを指定 oloyment_licensing.txt True   ロブイン(ユーザー固 の情報は読み取れたい







22 I WITTERのテーダ 家照	(1)(週11日回致上限について)	
Project1	×	通信回数が規定数を超過すると、エラーで しばらく(最大15分)接続できなくなる
[FireDAC][Phys][C reached for endpo	Data]The request limit of [15] has been int r.com/1.1/friends/ids.icon]. Please weit until	
2019-08-15T09:51	:42+09:00 to submit additional requests.	
- 75		
Twitter APIを利用し	たツイートのキーワート検索	
目的: ⑤入力したワードに関連する直 -************************************	ブロップートを検索 ****-*-******************************	
begin if (Trim(LabeledEdit1.Text) = '') raise Exception.Create('検索ワー	then - ドは必ず指定して下さい。`):	
// 対象データを抽出 FDQueryl.Close; FDQueryl.SQL.Clear: FDQueryl.SQL.Add(*SELECT Text, R	etweet_Count, Favorite_Count. 10'):	「素を
FDQueryI.SGL.Add('FROM Tweets') // 検索ワード(※SearchTermsを附旧 FDQueryI_SQL Add('WHERE SearchTe	: RE句に入れることで【ソース4】①とは検索条件が変わる) rms = ''' + Trim(LabeledEditl Text) + ''' ')	
if (StrToIntDof(LabeledEdit2.Text	.0) > 0) then // リツイート数の下限指定時	
FDOuery1.SQL.Add(' AND Retweet_ end; if (StyleIntDef()sheledEdit3_Text	Count >= ' + LabeledEdit2.Text):	
begin FDQuery1.SQL.Add(' AND Favorite	Count >= ' + LabeledEdit3.Text):	
end: if (CheckBox1.Checked) then // E	lah語のツィートに限定する時 50件ずつデータを取得	はするため、
FDOuery1. SOL. Add(' AND Lang = ' end:	·ja'' '): 「回の通信(50件のみ] データが得られない場	取得)で必要十分な 合は、増やす
FDOuery1.FetchOptions.RowsetSize FDOuery1.Open: // SOL実行 SetColWidths: // DBGridの列幅設定	:= 100: // 1回の通信で取得する件数 (※増やし通ぎるとレスホンズ) 通信回数制限の原因になる/	www.cap.co. とめ、注意)
end:		
23 Twitterのデータ参照	⑤(テーブル・ビューの一覧)	
<代表的なテーブルの一覧	>	
テーブル名	概要	
Favorites	・自分が「いいね」を送ったツイートを一覧表示	
Following	<ul> <li>・ 「いいね」の追加および解除</li> <li>・ 自会がステローレズいるユーザーの一覧書言など</li> </ul>	
Following	<ul> <li>・自分および自分がフォローしているユーザーの**</li> </ul>	、
Tweets	・WHERE句にSearchTermsを指定することで、特	定の検索ワードで検索
	・ツイートの作成と削除	
<代表的なビューの一覧>		
ビュー名	概要	
AccountSottings	自分のアカウントに関する設定を取得	
Accountactungs		

UL 1	120.00
AccountSettings	自分のアカウントに関する設定を取得
AdInsights	自分が発信した広告に対する視聴者などの情報を取得
Followers	自分または指定したユーザーをフォローしているユーザーのリストを取得
Mentions	自分が受けた最新のメンション(自分宛ての返信も含む)を取得
Retweets	自分のリツイートのリストを取得
Trends	日時や国や地域(WoelD)を指定してトレンドトピックのリスト(50件)を取得
TweetStream	Twitterを流れるパブリックデータ(全ての公開ツイート)を取得
Users	キーワードや名前を基にユーザーの一覧を取得

※ 自分 = ログインによって認証されたユーザーを指す。

※ 照会専用アカウントでログイン時はユーザーを指定すれば参照できるものと、参照不可のものがある。更新は不可。





E. For	m1										•	x	
						1					_	_	
	ORORNO	ORORST	ORCSCD	ORORDT	ORPRCD	ORPRNM		ORÇ	TY ORUN	IPR ORSH	HD1 ~		
•	1000001	7	10001	20170331	A0001	8/インチ液晶 新ノカニサキッ	at_29← .t.		2 48	000	-11		
-	1000005	Y	10001	20170331	C0002	新197井キッ   今白師法週	/ P 818		1 00	980	-11		
-	1000019		10001	20170405	30003	「田田助かり留」 「雷気ジャーポ	wh		10 14	800			
	1000028		10001	20170407	N0001	フルーツのど飴	; ;		5	98	1		
	1000091		10001	20180901	A0001	商品名			3	0			
	1000092		10001	20180803	A0001				5	0			
(顧習	<u> 客</u> コード=	10001	σ	)受注データを	をスプレッド	89-HC	<b>①</b> ①出力	]			* *		
ر کې Goo	<sup>り名=</sup> gleスプ	レッドシー	- <b>h</b> (5)(	出力結果	)								]
J» Goo ≡	<sup>り名=</sup> gleスプ 面 ス	レッドシ・ くプレ・	ート⑤() ッドミ	出力結果	)	() () () () () () () () () () () () () (	タワンロード スプレッ IBM i か 録定なし マ 7	ドシートが新 いら取得した		成 され、 が登録す	×n7		5
J> Goo	<sup>り名=</sup> gleスプ	レッドシ・ くプレッ 今E	-►⑤() אין ע	出力結果	)	。 検索 オーナー#	タワンロード スプレッ IBM i か 記定なし ▼	ドシートが新いら取得した	近規作 F データフ ヨ分)	成され、 が登録さ 田	<u>×</u> nī A <sup>2</sup> z		5
J» Goo	<sup>ク名=</sup> gleスプ	レッドシ・ 、プレッ 今日	ート⑤() ッドミ 1 レオ	出力結果 ノート ペート_2019	) 0902_11	() 検索 オーナー#	タワンロード IBM i か 記定なし マ		新規作 データ ヨ 分) 1:12	成され、 が登録さ 田	<u>م</u> رًz		5
J> Goo	少名= gleスプ 画 ス	レッドシ・ くプレ・ 今日 し ファ	ート⑤() ッドショ ユ ポート_ パイル 新	出力結果 ノート ペート_2019 _201909( 編集 表示	) 09902_11 D2_11111 挿入	(2) 大 オーナー邦 11118 18 表示形	タワンロード IBM i か 記定なし ▼	<b>ドシートが新いら取得した</b> 自分 アドオン へ	新規作 データ ヨ分) 1:12 ルプ	成され、 が登録さ 田 	<u>×れっ</u> 数秒		5
J» Goo	2名= gleスプ		ート⑤() ッドミ コ レオ ポート	出力結果 ノート ペート 201909( 重集 表示 100% マ	) 09902_111 D2_11111 挿入 業 %	(2) 検索 オーナー# III118 18 表示形	タワンロード IBM i か 記定なし マ ビ デフォルト	ドシートが新いら取得した 日分 アドオン へ	新規作所 データ7 目分) 1:12 ルプ □	成され、 が登録さ 田 	<u>きれて</u> 数秒	ている こ 前	 5 
J> Goo ≡		レッドシ・ プレ・ 今E	ート⑤() ッドミ コ マイル 新 マイル 新 マート	出力結果 シート ペート_2019 _201909( 編集 表示 100% マ	) 09902_11 D2_11111 挿入 ¥ %	(2) 検索 オーナー邦 IIII18 18 表示形 	タワンロード IBM i か 記定なし ▼ E ータ ツール デフォルト	ドシートが新 ら取得した 自分 アドオン へ	新規作 データフ ヨ分) 1:12 ルプ □ マ   B	成され、 が登録さ 田 	<u>×れつ</u> A <sup>2</sup> <u>A</u>	ている ご 前	
J> Goo ≡	少名= gleスプ 画 ス	レッドシ・ プレ・ 今日 日 日 日 日	-ト⑤() ッドミ オート_ オート_ マイル * -ド	出力結果 ノート ペート_2019 _2019090 編集 表示 100% マ	) 09902_11 D2_11111 挿入 ¥ %	(2) 検索 オーナー相 IIII18 表示形 	タワンロード スプレッ IBM i か 記定なし マ E -タ ツール デフォルト	<b>ドシートが新いら取得した</b> 自分 アドオン へ ー・10	新規作所 データ7 目分) 1:12 ルプ □ マ  B	成 さ た が 登録 ひ 部 一 、 、 が 登録 録 で 一 、 、 が 登録 録 こ 一 、 、 、 、 、 、 、 、 、 、 、 、 、	st AZ 数秒	ている こ 記 記	 5 
J» Goo	2名= gleスプ		-ト⑤() ッドミ -ド マイル 第 - ド	出力結果 ノート ペート_2019 _201909( 編集 表示 100% マ B 顧客名	) 09902_111 D2_11111 挿入 ¥ % 商品	(2) 検索 オーナー# 11118 18 表示形  名	タワンロード IBM i か 記定なし ▼ 「 デフォルト D 数量	ドシートが新 ら取得した 日分 アドオン ∧ □ ▼ 10	新規作所 データ7 目分) 1:12 ルプ □ ・ 日 日 日 日 日 日 日 日 日 日 日 日 日 日 月 日 月 日 月	成され、 が登録さ 田 	×れて A <sup>z</sup> A	ている ご 前 G	 ₹
J> Goo ≡	ク名= gleスプ 画 ス 「た 1 2	レッドシ・ プレ・ 今E	ート⑤() ッドミ レオ ポート マイル 新 マイル 新 マード	出力結果	) 09902_111 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕 〕	(2) 検索 オーナー邦 11118 18 表示形 -0- -0 -0 -0 -0 -0 -0 -0 -0 -	タワンロード       スプレッ IBM i か       ibm i か	ドシートが新 ら取得した 自分 アドオン へ	新規作 データ7 ヨ分) 1:12 ・ B 金額 00	成され、 が登録さ 田 	<u>×れて</u> A <sup>2</sup> <u>A</u>	ている ご 前 G	
J> Goo	ク名= gleスプ 回 ス 「「 「 「 」 1 2 3	レッドシ・ プレ・ 今日 日 夏客コー	-ト⑤() ッドミ レオ ポート マイル 第 -ド 10001 10001	出力結果	) 09902_11 D2_11111 挿入 ¥ % 田商電 3 7 田商電 新~	(2) 検索 オーナー指 11118 18 表示形 .0.4 4 4 4 4 4 4 	タワンロード       IBM i か       ip       シール       デフォルト       D       数量       2       10	<b>ドシートが新いら取得した</b> 自分 アドオン へ ー 10 単価 ・ 4800 り 290		成され、 が登録さ 田 		ている こ 記 前	
J» Goo	ク名= gleスプ 「「「「「」」 「「」」 「」 「」」 「」 「」 」 、 」 、 」 、 」 、 、 、 、 、 、 、 、 、 、 、 、 、	レッドシ・ プレ・ 今日 単名コ・ 顧客コ・	ート⑤() ッドミ ・ オート ・ マイル 第 ・ ・ ド 10001 10001	出力結果 ノート ペート_2019 _201909( 真生 100% マ B 顧客名 株式式会社吉に 株式式会社吉に 株式式会社吉に	) 09902_11 02_11111 挿入 業 % 田商電 3 7 田商電 4 田田商	(2) 検索 オーナー邦 11118 18 表示形 -0 に 品名 インチ液晶す (クラ丼キッ) 動洗濯機	タワンロード IBM i か 記定なし ▼	ドシートが ら取得した ヨ分 アドオン ∧ マドオン ∧ 10 ■ 単価 4800 290 998	新規作所 データ7 目分) 1:12 ・ B 金額 100 00		×れて A <sup>2</sup> Z	ている ご 前 G	 5 
J> Goo ≡	ク名= gleスプ 画 ス 「 「 「 「 、 、 、 、 、 、 、 、 、 、 、 、 、	レッドシ・ プレ・ 今日 レファ 夏客コー	ート⑤() レオ ポート マイル 新 マイル 新 マイル 新 マイル 新 10001 10001 10001	出力結果 、 ート ペート_2019 ペート_2019 201909( 編集式式会社吉 間株株式式会社吉 調株株式式会社吉 に 本 の の の の の の の の の の の の の	) 09902_11 D2_11111 挿入 業 % 田田商商電電電 新 子 目 田田商商電	(2) 検索 オーナー邦 11118 18 表示形 .0,1 ・ こ インチ液晶 イクラ丼キット 1動洗濯機 気ジャーポット	タワンロード スプレッ IBM i か 記定なし ▼ E テフォルト ア 数量 2 10 1 10	バシートが新 ら取得した	ft規作     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f     f	成され、 が登録さ 田 		ている ご 前 G	
J> Goo ≡	ク名= gleスプ 回 ス 「 た 1 2 3 4 5 6	レッドシ・ 、 プレ・ 今日 日 夏客コー	-ト⑤() ッドミ レオ ポート マイル # ・ア   -ド 4 10001 10001 10001 10001	出力結果 、 一 ト 、 ー ト_2019 、 ー ト_2019 2019090 編 100% 、 国際株株式式会社主言 の会社主言 の会社主言言	) 09902_11 09902_11 11 挿入 業 % 西裔高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高高	(2) 検索 オーナー邦 11118 18 表示形 .0,4 C 3名 インチ液晶 イクラ丼キット 動ジャーボット レーツのど飴	タワンロード スプレッ IBM i か 記定なし マ ド アフォルト の 数量 2 10 1 10 5	<b>ドシートが新 ら取得した</b> 自分 アドオン へ 10 単価 単価 4800 290 9980 1480			<u>され</u> A 文 型 型 型	ている こ 記 前 。	 3 
J> Goo ≡	ク名= gleスプ 「「「「」 「「」 「」 「」 「」 「」 「」 「」 「」	レッドシ・ 、 プレ・ 今日 して ファ 顧客コー	ート⑤( ソドミ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	出力結果 、ート ペート_2019 _201909( 重都式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式式	) 09902_11 02_11111 挿入 業 % 西商電 朝年 日田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田田	(2) 検索 オーナー邦 11118 18 18 18 18 18 18 18 18	タワンロード スプレッ IBM i か 記定なし ▼ デフォルト D 数量 2 10 5 3 5	ドシートが ら取得した で 日分 アドオン へ 10 単価 単価 1480 998 1480		求され、 が登録さ		ている ご 記 回 の	





	[FDConnection1]		
ドライバまたはオーバーライト	する接続定義の名前	を増択してから、パラメータをセットア	ツブします
定義 オブション 情報 50	QL スクリプト		① ドライバた
ドライバロ(2):	CDataGoogleDrive	N N	「CDataGoogleDrive II」 この この この この この この この に 設定
接続定義名(1):			
テストロ ウイ	ザード( <u>W</u> ) デフォル	いに戻す(化) ヘルプ(化)	
1153-2		值	77+11.5
DriverID		CDataGoogleDrive	CDataGoogleDrive
TeamDriveSupport			False
MaximumColumnSize		o 11.44 - 1	<ol> <li>⑦開発端末以外の端末で動作させる場合は</li> </ol>
MaxRows	(	④ 接続テスト	「deployment_licensing.txt」に記載された
Other			→ ランタイムキーを「RTK」に指定する必要がある
Readonly		False	
RTK		000000000000000000000000000000000000000	000 <del>00000000000000000000000000000000000</del>
SSIServerCert			
SupportEnhancedSQL		True	True
Initiate() Suth		60 GETANDREEDECH	60 DEF
OAuthAccessToken		GETANDREFRESH	
OAuthClientId			
OAuthClientSecret			③ InitiateOAuthを 「GETANDREFRESH」に設定
OAuthRefreshToken			
OAuthSettingsLocation		%APPDATA%##CData##Google	eurive uata Provide: %APPDATA%+*CData**GoogleDrive Data Provide
CAUGIVEINE			•
			OK TOUC
	/	·····································	
Google Drive3	(フルテキスト	検索)	
Google Drive③	(フルテキスト Q ドライブを検索	検索)	Ø ֎ Ⅲ GS
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ *	·検索) -	© ❀ Ⅲ GSL ■ 0
Google Drive③ トライブ + 新祝	(フルテキスト Q ドライブを検索 マイドライブ * ウイックアクセス	·検索) 	© ❀ Ⅲ GS. ■ (
Google Drive③ ▲ ドライブ + 新規 ② 栄精	(フルテキスト Q ドライブを検索 マイドライブ * クイックアクセス	·検索)	
Google Drive③ ドライブ + 新祝 ② 栄輔 ③ マイドライブ	(フルテキスト Q ドライブを検索 マイドライブ + クイックアクセス	·検索)	
Google Drive③ ドライブ + 新祝 ② 深端 ③ マイドライブ 二 For ③ 素質使用したアイ	(フルテキスト Q ドライブを検索 マイドライブ * クイックアクセス	·検索)	
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ ▼ クイックアクセス m1	·検索)	② ② 部 III G St 目 C レ名またはファイルの内容から検り込み 海海
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ ★ タイックアクセス m1	・検索) - - 覧の作成 - 日田和	② ⑧ Ⅲ GS 目 1)名非たはファイルの内容から続い込み 商事
Google Drive③ ドライブ + 新祝 ② 候補 ③ マイドライブ ※、共有アイテム ③ 未対史明したアイ ☆ スター付き 副 コミ酒	<ul> <li>(フルテキスト</li> <li>マイドライブ *</li> <li>ウイックアクセス</li> <li>m1</li> <li>①ファイル-</li> <li>Id</li> </ul>	·検索) - 覧の作成 - 日田和	② ⑧ Ⅲ G Si 目 い名またはファイルの内容から核り込み 商事 Name TeamDriveId ▲
Google Drive③ ドライブ + 新規 ② 候種 ③ マイドライブ 二 新売アイテム ③ 転送使用したアイ ☆ スター付き 副 コミ病	(フルテキスト Q ドライブを検索 マイドライブ ▼ クイックアクセス 「 10 1W_5mihaDDz3JCI 1Vrftz-KrS3M81№F	・検索) - 覧の(作成 HMA498yugNxz-at1DH13djvA 56Aer9-79C)YEoMa3v7275-5M	② ⑧ Ⅲ GS 目 い名またはファイルの内容から絞り込み 商事 Name eSTLZS レポート_20190902_111118 ME1xk レポート 20190902_114242
Google Drive③ ▲ ドライブ + 新規 ② 候補 ③ マイドライブ ③ ボビ共同したアイ ☆ スター付き 回 ゴミ酒	(フルテキスト Q ドライブを検索 マイドライブ ▼ クイックアクセス m1 ① ファイル- Id 1W_5mihaDDz33Ci 1Vrfrz-KrS3M8JNf5 1p6mf8PeSTOITgg	・検索) -覧の作成 HMA498yugNxz-at1DHI3djvA 56Aer9-79OYFgMg3xZ9Z5gYN yJLoyRXDKOvPOeUxmG9imWg	② ② ② 部 III G Si       目       レ名またはファイルの内容から絞り込み       商事       Name       reamDriveId       ● STLZs       レポート_20190902_111118       ME1xk       レポート_20190902_114242       pRZWj3y8       レポート_20190902_113946
Google Drive③ ドライブ + 転机 ② 候猫 ③ マイドライブ ③ マイドライブ ③ マイドライブ ③ 東近使用したアイ ☆ スター付き 副 コミ酒	<ul> <li>(フルテキスト</li> <li>マイドライブ *</li> <li>クイックアクセス</li> <li>m1</li> <li>① ファイル-</li> <li>Id</li> <li>1W_5mihaDDz3JCl</li> <li>1Vrfrz-KrS3M8JNf5</li> <li>1p6mf8PeSTOlTgq</li> <li>1myqTV5d9TE9z8</li> </ul>	・検索) ファイル 一覧の作成 HMA49ByugNxz-at1DHI3djvAi 56Aer9-79OYFgMg3x2925gYN IyLoyRXDKOvPOeUxmG9imWp RqHIGdmDVUyIzkmXurBqoI9HC	② ⑧ Ⅲ GS ■ 少名またはファイルの内容から絞り込み 寄事 Name ESTLZS レポート_20190902_11118 WE1xk レポート_20190902_114242 pRZWJ3y8 レポート_20190902_113946 CV9mYk レポート_20190830_214910
Google Drive③ ▲ ドライブ + 新規 ② 候構 ③ マイドライブ 器、共有アイテム ③ ポビ使用したアイ ☆ スター付き 回 コミ語	<ul> <li>(フルテキスト</li> <li>マイドライブ *</li> <li>クイックアクセス</li> <li>m1</li> <li>① ファイル-</li> <li>Id</li> <li>IVrfrz-KrS3M8JNff</li> <li>1p6mf8PeSTOITgq</li> <li>1myqTrV5d9TE92R</li> <li>1CcdOLRVbESgayr</li> </ul>	・検索) -覧の作成 HMA49ByugNxz-at1DHI3djvA 56Aer9-79OYEgMg3x29Z5gYR IyLoyRXDKOvPOeUxmG9imWp XqHiGdmDVUyIzkmXurBqoi9HC nuuHrtxrbWzIrTix7dIoBkQTsb	② ● 部 GS       ■       ▶24またはファイルの内容から絞り込み       協事       ■       ▶25125       レポート_20190902_11118       WE1xk       レポート_20190902_114242       pRZWj3y8       レポート_20190902_113946       CV9mYk       レポート_2019030_214910       wkf_I       テクニカルレポート2019その1
Google Drive③ ドライブ + 新規 ② 候補 ③ マイドライブ 二 ドライブ 二 ドロ 二 ドロ 二 ドロ 二 ドロ 二 ドロ 二 ドロ	(フルテキスト Q ドライブを検索 マイドライブ ▼ クイックアクセス I 10 1W_5mihaDD23JCI 1Vrfrz-KrS3M8JNf5 1p6mf8PeSTOITgq ImyqtTV5d9TE92R 1CcdOLRVbESgayr	・検索) ファイル 一覧の作成 HMA49ByugNxz-at1DHI3djvAi 56Aer9-79OYFgMg3x29Z5gYP iyLoyRXDKOvPOeUxmG9imWp &qHIGdmDVUyIzkmXurBqoI9HC nuuHrtxrbWzlrTix7dIoBkQTsb	② ● ※ (GS)       ■       レ名またはファイルの内容から絞り込み       審事       ■       Name       eSTLZs       レポート_20190902_11118       ME1xk       レポート_20190902_114242       pRZWJ3y8       レポート_20190830_214910       wkf_J       アクニカルレポート2019その1
Google Drive③ ドライブ + 新規 ② 栄潮 ③ マイドライブ ③ マイドライブ ③ ボビ使用したアイ ☆ スター付き 回 ゴミ語	<ul> <li>(フルテキスト</li> <li>マイドライブを検索</li> <li>マイドライブ *</li> <li>クィックアクセス</li> <li>m1</li> <li>① ファイル-</li> <li>Id</li> <li>1W_5mihaDDz3JCl</li> <li>1Vrfrz-KrS3M8JNf5</li> <li>1p6mf8PeSTOITgq</li> <li>1myqtTV5d9TE9zR</li> <li>1CcdOLRVbESgayr</li> </ul>	・検索) - 覧の作成 HMA49ByugNxz-at1DHI3djvA 56Aer9-79OYFgMg3xZ9Z5gYN IyLoyRXDKOvPOeUxmG9imWp tqHiGdmDVUyIzkmXurBqoI9HC nuuHrtxrbWzlrTix7dIoBkQTsb	② ② ② 部 III GS、       目       レ名またはファイルの内容から絞り込み       第事       レ名またはファイルの内容から絞り込み       第事       ESTLZs     レポート_20190902_111118       ME1xk     レポート_20190902_114242       pRZWj3y8     レポート_20190902_113946       CV9mYk     レポート_20190830_214910       wkf_I     テクニカルレポート2019その1
Google Drive③ ドライブ ・ 新規 ② 栄補 ③ マイドライブ ③ マイドライブ ③ ボビ使用したアイ ☆ スター付き 回 ゴミ語	<ul> <li>(フルテキスト</li> <li>マイドライブ *</li> <li>クイックアクセス</li> <li>m1</li> <li>① ファイル-</li> <li>Id</li> <li>1W_5mihaDDz3JCl</li> <li>1Vrfrz-KrS3M8JNf5</li> <li>1p6mf8PeSTOlTgq</li> <li>1myqtTV5d9TE9zR</li> <li>1CcdOLRVbESgayr</li> </ul>	・検索) ファイル 一覧の作成 HMA49ByugNxz-at1DHI3djvAi 56Aer9-79OYFgMg3xZ9Z5gYN IyLoyRXDKOvPOeUxmG9imWp RqHIGdmDVUyIzkmXurBqoI9HC nuuHrtxrbWzIrTix7dIoBkQTsb	② ② ② ③ Ⅲ GS         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □         □      <
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ * クイックアクセス 「 10 1W_5mihaDDZ3JCI 1Vrfrz-KrS3M8JNf5 1p6mf8PeSTOITgq 1myqtTV5d9TE92R 1CcdOLRVbESgayr	・検索) -覧の作成 HMA49ByugNxz-at1DH13djvA 56Aer9-79OYFgMg3x29Z5gYN IyLoyRXDKOvPOeUxmG9imWp 2qHiGdmDVUyIzkmXurBqol9HC nuuHrtxrbWzlrTix7dIoBkQTsb	② ② ② ② ② 第 GS         □ ② ③ 第 GS         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ □ ○ ○         □ □ □ □ ○ ○         □ □ □ ○ ○         □ □ □ ○ ○
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ ▼ クイックアクセス 「10 1W_5mihaDDz3JCI 1Vrfrz-KrS3M8JNfs 1p6mf8PeSTOITgq ImyqtTV5d9TE92R 1CcdOLRVbESgayr	・検索) ファイル 一覧の作成 HMA49ByugNxz-at1DHl3djvAi 56Aer9-79OYFgMg3x2925gYP iyLoyRXDKOvPOeUxmG9imWp kqHlGdmDVUyIzkmXurBqol9HC nuuHrtxrbWzlrTix7dIoBkQTsb	② ③ 部 #: GS         ■         ▶         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ●         ● <t< td=""></t<>
Google Drive③	(フルテキスト Q ドライブを検索 マイドライブ * クイックアクセス 「 10 1W_5mihaDDz3JCl 1Vrfrz-KrS3M8JNf5 1p6mf8PeSTOITgq 1myqtTV5d9TE92R 1CcdOLRVbESgayr	・検索) - 「 短の作成 ー覧の作成 ー覧の作成 ー 目の 本 の の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に 成 の に の の に の の に の の し し 引 る は の に の の し し 引 る は の に の で の い し し 引 る い ろ い る い ろ い る い ろ い ろ い る い ろ い て の い し い る い ろ い の い し い る い ろ い の い し い し い る い ろ い の い い し い る い の い い し い し い し い し い し い の い い い い い い い し い し い の い い い い い い い い い い い い い	② ③ 部 ::: GS:         ■         ▶28また(はファイルの内容から絞り込み 寄事         ▶28また(はファイルの内容から絞り込み (新事)         ● Name         ● ESTLZS         ● レポート_20190902_11118         ME1xk         ● レポート_20190902_114242         ● PSZWJ3y8         ● レポート_20190902_113946         CY9mYk       レポート_20190902_113946         CY9mYk       レポート_20190902_113946         CY9mYk       レポート_2019701         ■       ■         【図28】でセルにに「株式会社吉田商事」と セットしたこスプレッドシートがヒットした
Google Drive③ ドライブ + 新規 ② 栄補 ③ マイドライブ ③ ボビ使用したアイ ☆ スター付き 回 ゴミ語 ・	<ul> <li>(フルテキスト</li> <li>スドライブを検索</li> <li>マイドライブ *</li> <li>クィックアクセス</li> <li>m1</li> <li>① ファイル-</li> <li>Id</li> <li>1W_5mihaDDz3JCi</li> <li>1Vrfrz-KrS3M8JNf5</li> <li>1p6mf8PeSTOlTgq</li> <li>ImyqtTV5d9TE9zR</li> <li>1CcdOLRVbESgayr</li> </ul>	・検索) ファイル 古田和 HMA49ByugNxz-at1DH13djvA 56Aer9-79OYFgMg3xZ9Z5gYN IyLoyRXDKOvPOeUxmG9imWp kqHiGdmDVUyIzkmXurBqoI9HC nuuHrbxrbWzIrTix7dIoBkQTsb	② ・ ::: GS       B       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P       P        P

