IHS > Decision Support Tool

FieldDIRECT™ Data Services (FDDS)

October 2012

The Source for Critical Information and Insight™

www.ihs.com

FDDS
October 2012

IHS Inc.
15 Inverness Way East
Englewood, Colorado 80112
303-736-3000

Contents

What’s New (Version History)

Version 4.2.1.0 – October, 2012

No new FDDS URLs are introduced in this version. This release consists solely of one new feature and several bug fixes. Please note that the docWellTests retrieval issue described in item iii below introduces a behavorial change which may require modifications to your FDDS retrieval software. In addition, a new section has been added to this document which discusses how to maintain consistency between your FDDS retrievals and the Production Explorer database. You must detect and process server-side record deletions as described in the section entitled “How To Synchronize your local database with the FieldDIRECT™ database”.

i. New Feature: Prior to version 4.2.1.0, FDDS retrieved a maximum of 100,000 records for each data pull. This approach introduced some complexity in our customers’ retrieval software. To make sure all records have been retrieved, the customer needs to know the FDDS maximum count (i.e., 100,000). If a retrieval returns less than the maximum count, all current records have been received. If a retrieval returns exactly the maximum count, more records may or may not be available. To find out, a new request must be issued with a StampStartDate parameter equal to the stamp date of the last record retrieved. This approach works, but returns at least one duplicate record.

Version 4.2.1.0 includes a new feature which can be used to simplify FDDS retrieval software.   FDDS can return http response headers to indicate whether or not another retrieval is necessary, and, if it is, what to use for a StampStartDate parameter in the next retrieval. Retrievals are structured to guarantee that no duplicate records are received.

These new features are triggered by adding the 'MoreRecordsFlag=yes' parameter. Details are provided in the "FDDS URL" section of this document.

ii. Bug Fix: In version 4.2.0.0, the eqpEmissionControlDevice table retrieval fails unless you specify a table name of eqpEmissionControlDevices (note the trailing 's'). The Table4.2 URL will now work correctly with a tablename of either eqpEmissionControlDevice (the correct name), or eqpEmissionControlDevices.

iii. Bug Fix: Prior to this release, FDDS retrievals for the docWellTests table were missing four fields: ShutInTubingPressure, ShutInCasingPressure, FlowingTubingPressure, and FlowingCasingPressure. **The docWellTests dataset returned using this new version will include these four new fields. Therefore, your FDDS retrieval software may need to be modified to support this change.**

Version 4.2.0.0 – June, 2012

1) New Feature: Data pulls are available for both the v4.1 and v4.2 database schemas. To support this feature, several optional parameters have been added to the FDDS URL. Please reference "The FDDS URL" section in this document for more detail.

The new v4.2 tables, fields, and other schema changes listed in section *'4.2 schema changes'* of the "4.2.0.0 FDDS Release Notes" document can be downloaded.

2) fyi: If data has been entered into any of the new v4.2 tables or fields prior to the deployment of FDDS version 4.2.0.0, the 'stampstartdate' parameter must be backdated in order to retrieve this information. For example, if data was entered for Emission Control Devices (ECD), the FDDS pull for 'docEmissionControlDevices' data will be empty unless the 'stampstartdate' parameter is backdated to the date ECD data was first entered. It is not necessary, nor is it recommended, to backdate the 'stampstartdate' parameter to the project start date.

3) Bug Fix: FDDS no longer ignores the millisecond field in the projectstartdate and stampstartdate. As a result, there are fewer duplicate records when, after receiving a dataset with 100,000 records, another data set is requested.

4) Bug Fix: Prior to version v4.2, some of the more complex queries did not always sort the returned dataset correctly by 'Stamp'. The various sub queries were sorted correctly, but overall results were sometimes sorted incorrectly. Queries for the following tables were corrected in version 4.2.0.0: docChemicalReports, docCompressors, docDownTime, docEngineReports, docEquipFailure, docMonthlyElecReports, docPumpReports, docSealReports, docWellStatus, docWellTests, eqpCompressors, eqpElectricMotors, eqpEngines, eqpPumps, eqpTubing, ptsChemicalInjPt, and ptsChemicalReports.

5) Bug Fix: FDDS consumer processes will no longer see the occasional deadlock error. Potential deadlocks are now resolved on the FDDS server and are therefore transparent to the FDDS consumer.

Version 1.0.0.1 – January, 2012

1) Added two new parameters: xmldatasetname and xmltablename. Please reference the section entitled "The FDDS URL" in this document for operation details.
2) In May of 2011, the FieldDIRECT v4.1 database schema was updated to v4.2. Several new tables were added along with some minor structural changes to existing tables. Although version 1.0.0.1 of FDDS was designed to continue serving data using the 4.1 schema, introduction of the 4.2 schema caused some unanticipated FDDS behavioral changes. They are listed below, and fixed in FDDS version 4.2.0.0 when pulling data for v4.1:

   a) Even though the new 4.2 lstxxx tables were not included in the 4.1 schema, they could be retrieved using version 1.0.0.1. 'Table Not Found' errors should have been returned instead.
   b) Several beta users reported that their "eqpPumpingUnit" data pulls were different after the 4.2 database schema was introduced. The following new v4.2 fields were erroneously returned with version 1.0.0.1:
      i. PumpSize (float)
      ii. TheoreticalOutput (float)
   c) The 4.2 schema changed the type of the docPumpingUnits.StrokeSpeed field from integer to real. Version 1.0.0.1 of FDDS erroneously returned a real number instead of an integer.

Version 1.0.0.0 – April, 2011

Initial Release

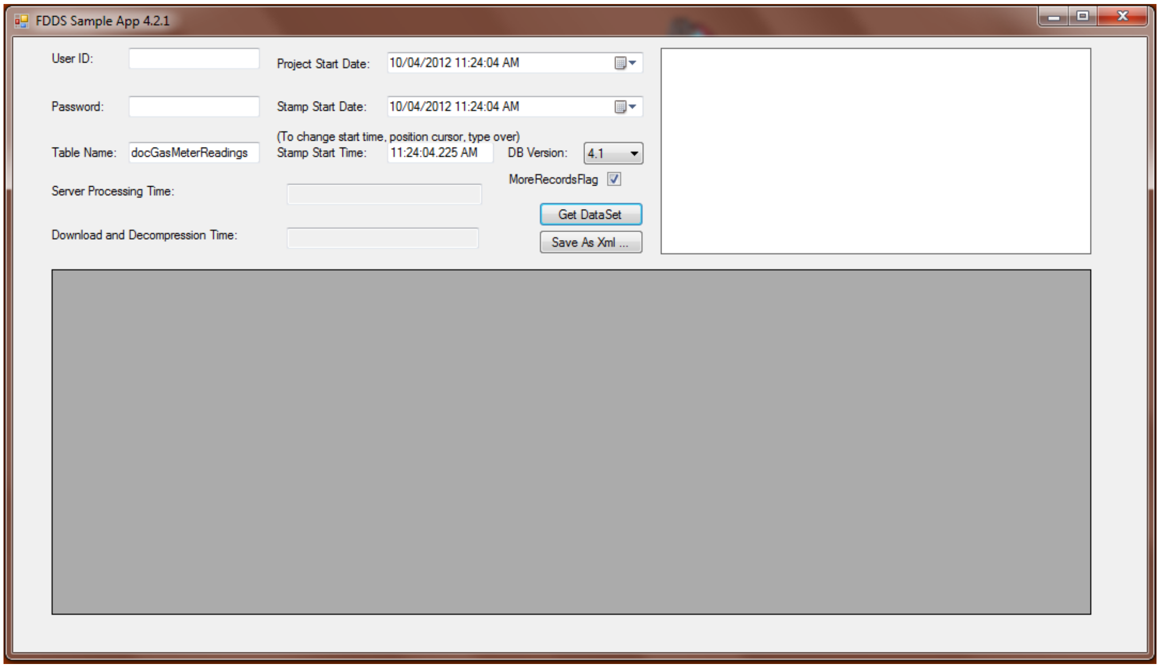
FDDS Introduction

FDDS, an acronym for FieldDIRECT™ Data Services, is a web service which can be used to retrieve your proprietary data residing on our FieldDIRECT™ database servers. Production Explorer (PE), another FieldDIRECT™ component, will continue to offer the same data retrieval services you have used in the past. In addition, PE offers many different report options to view your data. FDDS is available for those of you who want more control over your data pulls. You can, for example, create a background program to request daily updates for one or more tables. You can then transfer this data to your proprietary database. FDDS is a data retrieval service only. At this time, it does not offer data entry or data update services. Specific FDDS features include:

- FDDS is a web service hosted on our FieldDIRECT™ servers.
- FDDS allows you to retrieve data from more than 100 database tables.
- An FDDS request is initiated by issuing a URL (Uniform Resource Locator) over an existing internet connection.
- All internet data is transmitted over https using SSL (Secure Socket Layer). All data to and from our FDDS server is therefore encrypted and secure.
- Retrieved data is compressed and sent over an internet connection
- FDDS is intended to be called programmatically. In order to use FDDS effectively, you will need to develop software to issue FDDS requests and to process the returned data. To get you started, an FDDS sample application is available for download *(please refer to section 'FDDS Sample Application' page 6).*
- FDDS delivers data as a compressed data set. The FDDS sample application includes sample code to decompress and display this dataset. Sample code is also provided to save the dataset as an xml file.

FDDS Sample Application

An FDDS data retrieval request is initiated by issuing a URL. Although you can issue this URL from an internet browser, the data will be returned as a compressed dataset. Most browsers will just prompt you to save this data as a file. The resulting file is not readable. FDDS is therefore designed to be used within a software application. The FDDS Sample Application provides sample code to retrieve data by issuing an FDDS URL. The Sample App then decompresses the returned dataset and displays it in the bottom grid. When the FDDS Sample App is executed, you will be presented with the following dialog. Error and status messages are displayed in the upper right web control. The dialog's input controls are explained in section '*FDDS Sample App Control Fields' page 7.*



User ID:

Password:

Table Name: docGasMeterReadings

Server Processing Time:

Download and Decompression Time:

Project Start Date: 10/04/2012 11:24:04 AM

Stamp Start Date: 10/04/2012 11:24:04 AM

(To change start time, position cursor, type over)
Stamp Start Time: 11:24:04.225 AM

DB Version: 4.1

MoreRecordsFlag ☑

FDDS Sample App Control Fields

Each of the user input fields is collected and used to create the FDDS URL described in *section 'FDDS URL' page 11*. For example, the "User ID" input field corresponds to the URL's PARM_USERID parameter.

1. **User ID -** A valid FieldDIRECT™ userid. In general, records will only be retrieved if they "belong" to a property or well site which has been assigned to this userid. Property and well assignments can be established by your FieldDIRECT™ administrator.

   Example 1: You have requested tank data records from the docTankGauges table. A tank data record is retrieved only if the corresponding tank is located on a property or well site which is assigned to this User ID.

   Example 2: You have requested a list of engine definition records from the eqpEngine table. An engine definition record is retrieved only if the corresponding engine is on a property or well site assigned to this User ID.

2. **Password -** A valid password for the User ID specified.

3. **Table Name -** One of the FieldDIRECT™ table names listed in *'Tables Accessible via FDDS' page 17*. FDDS will return records from this table.

4. **Project Start Date -** This parameter is needed when retrieving data from any of the "doc" tables, or when requesting sumDailyProduction or sumDailyInjection data. Only records with docDates on or after the project start date will be included in the data pull. Each record's docDate field is entered during data entry and is intended to reflect the date and time data in that record was measured, not the date and time the data was entered using the data entry software. For tanks, this is

the date and time the tank was gauged. For meters, this is the date and time the meter reading was taken. For other entities, this is the date and time the applicable measurement for that entity was taken.

5. Not all table queries use the project start date. Check the table list in *section 'Tables Accessible via FDDS' page 17* to determine which tables require the project start date. (Beta Note: Even if a table does not require the Project Start Date, the beta version of FDDS will throw an error if a valid project start date is not provided) Note: *Section 'Using the FDDS URL – some examples' page 15* presents several examples which further explain how the project start date is used.

6. **Stamp Start Date -** This field corresponds to the date and time when a record was last modified. Only records modified on or after the stamp start date will be included in the data pull. For data entry records in the "doc" tables, this may or may not be the original insertion date. A record's initial stamp date corresponds to the date and time the record was first created. If a pumper corrects an existing record, that record's stamp date will change. Not all table queries use the stamp start date. Check the table list in *section 'Tables Accessible via FDDS' page 17* to determine which tables require the stamp start date. (Beta Note: Even if a table does not require the stamp start date, the beta version of FDDS will throw an error if a valid stamp start date is not provided). Note: *Section 'Using the FDDS URL – some examples' page 15* presents several examples which further explain how the stamp start date is used.

7. **Stamp Start Time -** This field is a continuation of the Stamp Start Date field and allows you to specify the stamp start date down to the millisecond level.

8. **DB Version -** This field allows you to specify the FieldDIRECT database version. FDDS maintains separate table queries for each

database version. When you request a table retrieval, FDDS will use the version query specified by this parameter.

9. **MoreRecordsFlag** – If checked, two http response headers will be returned to assist in requesting the next batch of records in your current data pull. See the section entitled "The FDDS URL" for more information.

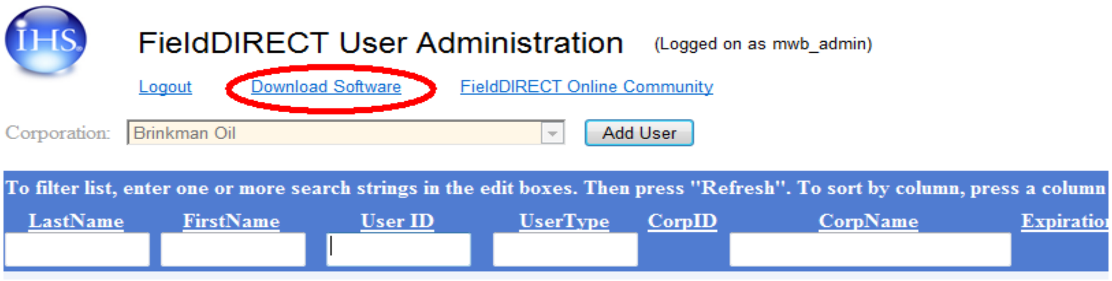
10. **Get Data Set Button -** When this button is pressed, an FDDS request URL will be sent to the FDDS service. The FDDS Sample Application will decompress the returned data set and display it in the grid.

11. **Save As Xml Button -** Allows you to save the data set displayed in the grid as an xml file.

How to download the FDDS Sample Application

The FieldDIRECT™ User Administration web application will allow you to download a zipped version of the FDDS Sample Application project. Note that in order to successfully download FDDS using the following steps, your userid must first be given permission to perform the download. If you cannot complete step 3 below, please call Customer Care to add the necessary permissions to your userid. **Customer Care's phone number is 1-800-IHS-CARE (1-800-447-2273).**

Step 1: Log on to FieldDIRECT™ User Administration and click the Download Software link……..



FieldDIRECT User Administration (Logged on as mwb_admin)

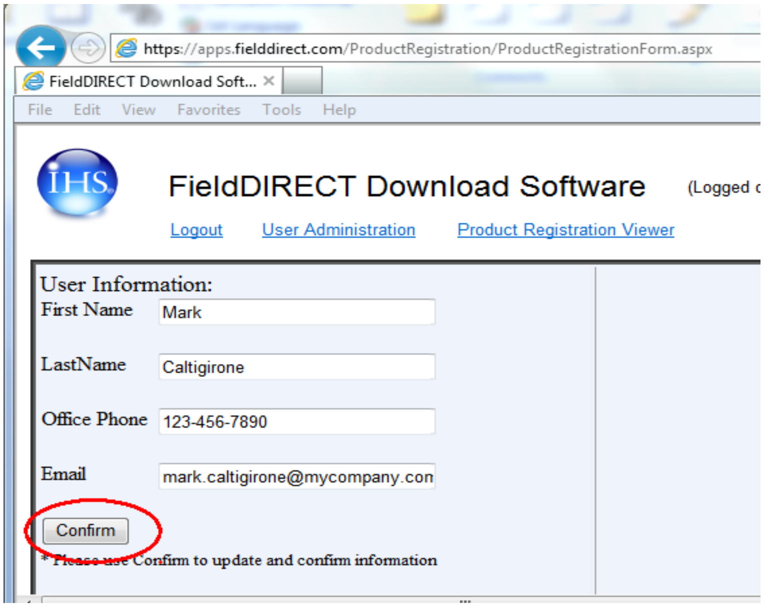
Logout Download Software FieldDIRECT Online Community

Corporation: Brinkman Oil

To filter list, enter one or more search strings in the edit boxes. Then press "Refresh". To sort by column, press a column

LastName FirstName User ID UserType CorpID CorpName Expiratio

Step 2: Press the Confirm button.



https://apps.fielddirect.com/ProductRegistration/ProductRegistrationForm.aspx

FieldDIRECT Download Soft...
File Edit View Favorites Tools Help

FieldDIRECT Download Software (Logged o

Logout User Administration Product Registration Viewer

User Information:

First Name: Mark

LastName: Caltigirone

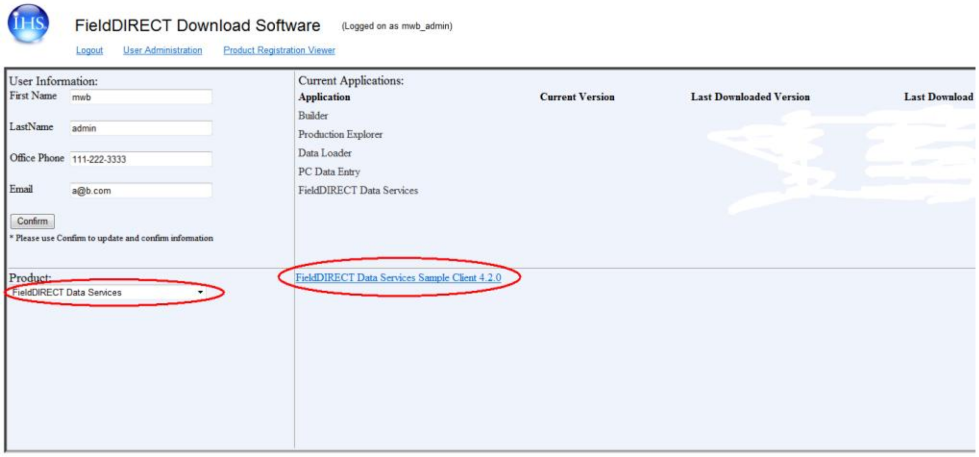
Office Phone: 123-456-7890

Email: mark.caltigirone@mycompany.con

* Please use Confirm to update and confirm information

Step 3: Choose “FieldDIRECT Data Services” from the Product dropdown.

- Click on the “FieldDIRECT Data Services Sample Client” link and follow the instructions



FieldDIRECT Download Software (Logged on as mwb_admin)

User Information:

First Name	mwb
LastName	admin
Office Phone	111-222-3333
Email	a@b.com

* Please use Confirm to update and confirm information

Product: FieldDIRECT Data Services

Current Applications:

Application	Current Version	Last Downloaded Version	Last Download
Builder			
Production Explorer			
Data Loader			
PC Data Entry			
FieldDIRECT Data Services			

FieldDIRECT Data Services Sample Client 4.2.0

FDDS Sample App System Requirements

1. The FDDS Sample App is packaged as a .Net C# Windows Forms Application project. You can load and execute the project using Microsoft Visual Studio 2008 or above. Like any other Visual Studio project, you can set breakpoints, step through the code, etc.
2. Your system must also be connected to the internet.

The FDDS URL

Data Retrieval Overview

In order to retrieve data from FDDS, you must issue a URL over an existing internet connection. This section describes the FDDS URL format.

Detailed explanation of URL syntax

Parameters enclosed in square brackets [] are optional. The brackets are informational only. They are not part of the URL.

https://apps.FieldDirect.com/FDDS/Table[PARM_DBVERSION]?userid=PARM_USERID&password=PARM_PASSWORD&tablename=PARM_TABLENAME&stampstartdate=PARM_STAMPSTARTDATE&projectstartdate=PARM_PROJECTSTARTDATE[&xmldatasetname=PARM_XMLDATASETNAME&xmltablename=PARM_XMLTABLENAME][&MoreRecordsFlag=PARM_MORERECORDSFLAG]

The above URL template is referenced in the following parameter definitions:

PARM_DBVERSION: An optional parameter which specifies the database version of the table you wish to retrieve. (4.1 Or 4.2). Defaults to 4.1. Examples:

Example 1: Retrieve data using the 4.2 schema of the database:
https://apps.fielddirect.com/FDDS/Table4.2?userid=...........

Example 2: Retrieve data using the 4.1 schema of the database (uses explicit parameter):
https://apps.fielddirect.com/FDDS/Table4.1?userid=.............

Example 3: Retrieve data using the 4.1 schema of the database (parameter defaults to 4.1):
https://apps.fielddirect.com/FDDS/Table?userid=..........

PARM_USERID: A valid FieldDIRECT™ userid. In general, records will only be retrieved if they "belong" to a property or well site which has been assigned to this userid. Property and well assignments can be established by your FieldDIRECT™ administrator.

Example 1: You have requested tank data records from the docTankGauges table. A tank data record is retrieved only if the corresponding tank is located on a property or well site which is assigned to PARM_USERID.
Example 2: You have requested a list of engine definition records from the eqpEngine table. An engine definition record is retrieved only if the corresponding engine is on a property or well site assigned to USERID.

PARM_PASSWORD: A valid password for the userid specified.

PARM_TABLENAME: One of the FieldDIRECT™ table names listed in *section 'Tables Accessible via FDDS' page 17*. FDDS will return records from this table.

PARM_STAMPSTARTDATE: This field corresponds to the date and time when a record was last modified. You may specify the date/time down to the millisecond level. Only records modified on or after the stamp start date will be included in the data pull. For data entry records in the "doc" tables, this may or may not be the original insertion date. A record's initial stamp date corresponds to the date and time the record was first created. If a pumper corrects an existing record, that record's stamp date will change. Not all table queries use the stamp start date. Check the table list in s*ection 'Tables Accessible via FDDS 'page 17* to determine which tables require the stamp start date. (Beta Note: Even if a table does not require the stamp start date, the beta version of FDDS will throw an error if a valid stamp start date is not provided). Note: '*Section Using the FDDS URL – some examples' page 15* presents several examples which further explain how the project start date is used.

PARM_PROJECTSTARTDATE:
Background Information: A FieldDIRECT project corresponds to all of the entities and their associated data that have been assigned to a ProjectID/UserID. When retrieving data via FDDS, the project start date should be set once for the project and then used in all subsequent calls to the service. Suppose Company A started using FieldDIRECT in September 15, 2010 on a trial basis and then rolled it out to the entire company on March 1, 2011. For their implementation of FDDS, Company A is only interested in the data which was gathered once they had the entire company on FieldDIRECT. In that case the project start date would be 3/1/2011 and would be used in all calls to the FDDS service. This would ensure that no data entered for dates (docDates) prior to 3/1/2011 would be retrieved, even though the company would have at least some data in the system prior to 3/1/2011. Also, even though data for days prior to 3/1/2011 may have been entered or edited (stamped) on or after 3/1/2011 that data is not retrieved if the project start date is set for 3/1/2011.

The PARM_PROJECTSTARTDATE parameter is therefore needed when retrieving project data from any of the “doc” tables, or when requesting sumDailyProduction or sumDailyInjection data.   Only records with docDates on or after the project start date will be included in the data pull.   Each record’s docDate field is entered during data entry and is intended to reflect the date and time data in that record was measured, not the date and time the data was entered using the data entry software. For tanks, this is the date and time the tank was gauged. For meters, this is the date and time the meter reading was taken. For other entities, this is the date and time the applicable measurement for that entity was taken. Although you may specify the project start date/time down to the millisecond level, our data entry programs truncate the milliseconds from the docDate fields.

Not all table queries use the project start date. Check the table list in *section 'Tables Accessible via FDDS' – page 17* to determine which tables require the project start date. (Beta Note: Even if a table does not require the Project Start Date, the beta version of FDDS will throw an error if a valid project start date is not provided). Note: *Section 'Using the FDDS URL – some examples' page 15* presents several examples which further explain how the project start date is used.

PARM_XMLDATASETNAME: Optional parameter. After the URL is issued, data is returned in a dataset. If this parameter is specified in the URL, the returned dataset will be tagged with the name designated by this parameter. Otherwise, the dataset name will default to ds_tablename, where tablename is designated by PARM_TABLENAME.

PARM_XMLTABLENAME: Optional parameter. After the URL is issued, data is returned in a dataset. The requested table is embedded in the dataset. If this parameter is specified in the URL, the embedded table will be tagged with the name designated by this parameter. Otherwise, the table name will default to tablename, where tablename is designated by PARM_TABLENAME.

PARM_MORERECORDSFLAG: MoreRecordsFlag is an optional parameter introduced in version 4.2.1.0. It can be set to 'MoreRecordsFlag=Yes' or 'MoreRecordsFlag=No'. If the parameter is not specified, it defaults to 'MoreRecordsFlag=No'.

If 'MoreRecordsFlag=No', the service operates exactly as it did prior to the release of version 4.2.1.0. A maximum of 100,000 records are returned, and the FDDS consumer is responsible for determining the next StampStartDate parameter. We suggest that the FDDS consumer should determine the last Stamp date in the batch of 100,000 records, and use that date as the StampStartDate parameter for the next data pull. At least one duplicate record will be returned using this method, but you are guaranteed to receive all records on the server. You should continue requesting records in this manner until FDDS recturns less than 100,000 records. At this point, you have retrieved all records which are currently available on the server.

If ‘MoreRecordsFlag=Yes’, FDDS will return two http response headers along with the retrieved data set. These two headers will allow the FDDS consumer to retrieve all records currently on the server without needing to know a maximum record count. All pulled records will be unique, so you won’t have to worry about duplicate records anymore. FDDS will return an indeterminate number of data set records along with the two http headers described below. **In future releases, this will be the default functionality. The hard-coded maximum record count of 100,000 will be deprecated in a future release. For that reason, we recommend that you start using this option as soon as possible.** Each response header is a name/string value pair. The first header is named ‘MoreRecords’ and will have the value of ‘Yes’ or ‘No’. If ‘Yes’, more records are available on the server. If ‘No’, you have retrieved all available records for this dataset. The second parameter is named ‘NextStampDate’. It will either be set to ‘None’ or will be set to a valid date/time string. All valid combinations of the two headers are listed below. If a combination is not listed, it will never happen…

(1) MoreRecords=Yes, NextStampDate=valid date time string. More records are available on the server. Use the NextStampDate string as your StampStartDate parameter for your next data pull. The next batch of records retrieved will not contain any duplicates. Note that this combination of response header values will be returned only if the requested table retrieval uses the StampStartDate parameter. See the “Tables Accessible via FDDS” section for a complete list of these tables.
(2) MoreRecords=No, NextStampDate=valid date time string. All available records have been retrieved (even if the pulled dataset is empty), so you don’t need to reissue your fdds url immediately. When new records do arrive on the server, you can retrieve them

using the NextStampDate as your StampStartDate parameter. Note that this combination of response header values will be returned only if the requested table retrieval uses the StampStartDate parameter. See the “Tables Accessible via FDDS” section for a complete list of these tables.

(3) MoreRecords=No, NextStampDate=None. This combination of response header values is returned only if the requested table retrieval does not use the StampStartDate parameter. See the “Tables Accessible via FDDS” section for a complete list of these tables.

Note: The latest version of our ‘FDDS Sample Application’ program includes sample code to retrieve the new http Response headers.

Using the FDDS URL – some examples

An Introductory Example

https://apps.FieldDirect.com/FDDS/Table?userid=pumper123&password=pumperpassword&tablename=docTankGauges&stampstartdate=2/1/2011 02:12:14.345 AM&projectstartdate=1/1/2009&MoreRecordsFlag=Yes

The URL listed above could be used to satisfy the following requirements:

- Retrieve tank gauge data for tanks serviced by a pumper with userid=pumper123. Pumper’s password is ‘pumperpassword’.
- Retrieve any tank gauge data which has been newly entered or edited on or after 2/1/2011 02:12:14.345 AM. This date is known to FDDS as the stamp start date since FieldDIRECT™ data

records include a datetime stamp field which is updated when a record is initially written or updated.

- Limit the data retrieved to tank data which was gauged/read after 1/1/2010 (project start date). This project start date corresponds to the docDate field in the docTankGauges table.
- Return the following http headers to assist in retrieving the next batch of records: MoreRecordsFlag and NextStampDate.

How to download all your data – an example

Data Retrieval Requirements**.** This section presents an example with the following requirements:

1. Retrieve all gas meter readings collected by "pumper123".
2. Retrieve all data collected on or after 3/1/2010.

Retrieval Steps.

1. Issue the following URL to request all gas meter records collected on or after 3/1/2010 (the project start date). Make sure the stamp start date is identical to the project start date. FDDS will return an indeterminate number of records. Each record will have been collected on or after 3/1/2010 (the project start date), and stamped on or after 3/1/2010 (the stamp start date). . https://apps.FieldDirect.com/FDDS/Table?userid=pumper123&password=pumperpassword&tablename=docGasMeterReadings&stampstartdate=3/1/2010&projectstartdate=3/1/2010&MoreRecordsFlag=Yes.
2. (Repeat this step until the http header named 'MoreRecords' is set to 'No'). The returned dataset is sorted by the Stamp field. If the returned http header named 'MoreRecords' is set to 'No',

you're done. There are no more records. Save the value of the http header named 'NextStampDate' so you can retrieve the next set of data later. Otherwise, if the returned http header named 'MoreRecords' is set to 'Yes', issue a new URL identical to the one just issued with the following exception: For the stamp start date parameter, use thevalue of the returned http header named 'NextStampDate'. It may be, for example, 6/13/2010 12:32:04.429 PM. The returned dataset is guaranteed to have unique records; none of the records will be duplicates of any previously returned records. . https://apps.FieldDirect.com/FDDS/Table?userid=pumper123&password=pumperpassword&tablename=docGasMeterReadings&stampstartdate=6/13/2010 12:32:04 .429 PM&projectstartdate=3/1/2010.

How to Synchronize your local database with the FieldDIRECT™ database

Keeping your local copy of the data tables, like sumDailyProduction and docGasMeterReadings, up to date is a bit more complicated that merely issuing successive FDDS retrieval calls with different StampStartDate parameters. One example of where the data you retrieve will differ is when a user updates a record. You'll get different results depending on whether the user deleted the old record and inserted a new record or simply edited the existing record. Also, the sumDailyProduction table is a derived (calculated) table dependent on source (measurement) data from other tables (i.e. you can't directly enter production data in FieldDIRECT™). This section presents a scenario which will explain how FieldDIRECT™ populates data tables, including the sumDailyProduction table, and the steps you need to take to ensure that your local copy of the tables accurately reflects the equivalent server side tables.

1. A pumper enters source/measurement data gathered on 5/15/2012. However, he mistypes the date and enters 5/1/2012. In our scenario let’s assume he’s entered a gas meter reading. When this is sent to the FD server, a new record is created in docGasMeterReadings and, upon insertion of the new record, an allocation request is scheduled for the property on which this meter exists for the date of 5/1/2012. Allocation requests are processed by an internal FieldDIRECT service called the Allocation Engine which runs on one of our backend FieldDIRECT servers. When the allocation request is processed by our Allocation Engine, a production record is inserted or updated in the sumDailyProduction table. This record will have 5/1/2012 as the docDate. In FD there are myriad ways to configure you properties so it is possible that this meter reading volume needs to be allocated to multiple production points. If so, a new record would be inserted/updated in sumDailyProduction for each production point. You could also have a case where multiple meter reading volumes need to be summed and allocated to a single production point. For now, let’s focus on a single gas meter and a single production point! ☺
2. When the data is downloaded via FDDS two records of interest are received. First is the new record in docGasMeterReadings. Second is the new record in sumDailyProduction. In both cases, your software checks to see if a record already exists based on the primary keys of each table. Since, in our example, it does not, a new record is inserted into each table for 5/1/2012.
3. Later, the pumper realizes his mistake and goes back and corrects the date from 5/1/2012 to 5/15/2012. Here is where things get interesting! The pumper can actually do this one of two ways. He can delete the old reading and insert a new one or he can update the existing reading. We’ll follow both as they lead to the same results in sumDailyProuduction but different results in docGasMeterReadings.
    a. If the old reading is deleted and a new reading is entered, that is exactly what happens on the server. The record containing the reading for 5/1/2012 is deleted from docGasMeterReadings and upon deletion an allocation request is scheduled for 5/1/2012

because we've had a measurement change on this date. At the same time a record is inserted into the fd_sysDeletes table. This record identifies the docGasMeterReadings record just deleted. In general, this table contains information (table name, key names and key values) that you will need to delete a record from a given table on your local machine in order to keep it in sync with the table on the server. After processing the deleted record, the new record for the volume reading on 5/15/2012 is inserted and an allocation request is scheduled for 5/15/2012. When the allocation request is processed for 5/1/2012, the Allocation Engine detects there is no longer any source/measurement data for that date so the 5/1/2012 record for that production point in sumDailyProduction is deleted. Once again a record is written to fd_sysDeletes providing the information that you will need to delete the record from your local sumDailyProduction table. Now the Allocation Engine processes the 5/15/2012 request and inserts a record into sumDailyProduction for this production point with a docdate of 5/15/2012.

   b. In this case, the pumper edited the reading and the existing record in docGasMeterReadings is simply updated to reflect the change in the docDate from 5/1/2012 to 5/15/2012. However, an allocation request is scheduled for both dates as we have a change in production on both dates. The allocation process would proceed as described in 3a.

4. Again, data is downloaded via FDDS. The records of interest downloaded will be different depending on the scenario. We'll follow each.
   a. A new record is received in the request for docGasMeterReadings data. This record will contain the measurement data for 5/15/2012. A new record is also received for the sumDailyProduction table with a docdate of 5/15/2012. Two new records are received when retrieving the data for the fd_sysDeletes table. One record contains the data (table name and key information) you will need to delete the appropriate record in the docGasMeterReadings and the other will contain the data you will need to delete the appropriate record

in the sumDailyProduction table. Once the records retrieved from the fd_sysDeletes table have been processed, there will be no data on 5/1/2012 in either the docGasMeterReadings or the sumDailyProduction table.

b. A record is received in the request for docGasMeterReadings containing the measurement data which now has the date of 5/15/2012. You can update your local database with the new, edited date by matching the primary key on the just received data record with the primary key of the existing record in the docGasMeterReadings table and performing at update. A record is also received when the request for sumDailyProduction data is fulfilled. This record, containing the production data on 5/15/2012, will not exist in your local database and can be inserted. One new record is received when retrieving the data for the fd_sysDeletes table. This record will contain the data (table name and key values) you will need to delete the 5/1/2012 record in the sumDailyProduction table. You will find a record in the sumDailyProduction table in your local database with this key (EntityID, EntityTypeID, docDate) and it can safely be deleted. Once the record retrieved from the fd_sysDeletes table has been processed, there will be no data on 5/1/2012 in either the docGasMeterReadings (because the date of the existing record has been edited) or the sumDailyProduction table (because the record has been deleted).

Summary: In addition to retrieving new and updated table records, you will need to process the table containing deleted record information (fd_sysDeletes) in order to keep your local copy of the database synchronized with the FD server.

Tables accessible via FDDS

TableName	Available DB Versions	query uses stampstartdate parameter	query uses projectstartdate parameter	query sort fields	primary key
docAutomationDevice	4.1 - 4.2	✓	✓	stamp	docIDAutomationDevice
docChemicalReports	4.1 - 4.2	✓	✓	stamp	docIDChemicalReports
docCompressors	4.1 - 4.2	✓	✓	stamp	docIDCompressors
docDownTime	4.1 - 4.2	✓	✓	stamp	docIDDownTime
docElecSubPumps	4.1 - 4.2	✓	✓	stamp	docIDElecSubPumps
docEmissionControlDevices	4.2	✓	✓	stamp	docIDEmissionControlDevices
docEngineReports	4.1 - 4.2	✓	✓	stamp	docIDEngineReports
docEquipFailure	4.1 - 4.2	✓	✓	stamp	docIDEquipFailure
docFluidLevels	4.1 - 4.2	✓	✓	stamp	docIDFluidLevels
docGasDispositions	4.1 - 4.2	✓	✓	stamp	docIDGasDispositions
docGasMeterReadings	4.1 - 4.2	✓	✓	stamp	docIDGasMeterReadings
docInjectionReports	4.1 - 4.2	✓	✓	stamp	docIDInjectionReports
docLactTickets	4.1 - 4.2	✓	✓	stamp	docIDLactTickets
docLiquidMeterReadings	4.1 - 4.2	✓	✓	stamp	docIDLiquidMeterReadings
docLiquidTransfer	4.1 - 4.2	✓	✓	stamp	docIDLiquidTransfer
docMonthlyElecReports	4.1 - 4.2	✓	✓	stamp	docIDMonthlyElecReports
docOffshoreReports	4.1 - 4.2	✓	✓	stamp	docIDOffshoreReports
docOilDispositions	4.1 - 4.2	✓	✓	stamp	docIDOilDispositions
docPlungerLift	4.1 - 4.2	✓	✓	stamp	docIDPlungerLift

docPumpingUnits	4.1 - 4.2	✓	✓	stamp	docIDPumpingUnits
docPumpReports	4.1 - 4.2	✓	✓	stamp	docIDPumpReports
docRunTickets	4.1 - 4.2	✓	✓	stamp	docIDRunTickets
docSealReports	4.1 - 4.2	✓	✓	stamp	docIDSealReports
docTankGauges	4.1 - 4.2	✓	✓	stamp	docIDTankGauges
docWaterDisposition	4.1 - 4.2	✓	✓	stamp	docIDWaterDisposition
docWellHeadPressures	4.1 - 4.2	✓	✓	stamp	docIDWellHeadPressures
docWellStatus	4.1 - 4.2	✓	✓	stamp	docIDWellStatus
docWellTests	4.1 - 4.2	✓	✓	stamp	docIDWellTests
eqpAutomationDevice	4.1 - 4.2	✓		PropertyID	AutomationDeviceID
eqpCompressors	4.1 - 4.2	✓		stamp	CompressorID
eqpECDSource	4.2	✓		Stamp	ECDSourceID
eqpElecSubPumps	4.1 - 4.2	✓		stamp	ElecSubPumpID
eqpElectricMotors	4.1 - 4.2	✓		stamp	ElectricMotorID
eqpEmissionControlDevices	4.2	✓		stamp	ECDID
eqpEngines	4.1 - 4.2	✓		stamp	EngineID
eqpPlungerLift	4.1 - 4.2	✓		stamp	PlungerLiftID
eqpPumpingUnits	4.1 - 4.2	✓		stamp	PumpingUnitID
eqpPumps	4.1 - 4.2	✓		stamp	PumpID
eqpRodPump	4.1 - 4.2	✓		stamp	RodPumpID
eqpRods	4.1 - 4.2	✓		stamp	RodID
eqpTank	4.1 - 4.2	✓		stamp	CompID
eqpTankStrapDetails	4.1 - 4.2			StrapID, StrapOrder, Height	StrapID, StrapOrder

eqpTankStrapMaster	4.1 - 4.2	✓		TankID, StrapDate, StrapID	StrapID
eqpTubing	4.1 - 4.2	✓		stamp	TubingID
fd_sysCorpLabels	4.1 - 4.2			n/a	CorpID, EntityTypeID
fd_sysCorporations	4.1 - 4.2			n/a	CorpID
fd_sysDataFieldLabels_ECD	4.2			n/a	CorpID, ECDEntityTypeID
fd_sysDeletedEntities	4.1 - 4.2	✓		entryStamp	EntityID, EntityTypeID
fd_sysDeletes	4.1 - 4.2	✓		entryStamp	
fd_sysGatheringSite	4.1 - 4.2	✓		PropertyID, SiteName	GatheringSiteID
fd_sysProperties	4.1 - 4.2	✓		PropertyName	PropertyID
fd_sysStream	4.1 - 4.2	✓		stamp	StreamID
fd_sysUserProperties	4.1 - 4.2			PropertyName	UserID, PropertyID
fd_sysUsers	4.1 - 4.2			n/a	UserID
fd_sysWellSite	4.1 - 4.2	✓		PropertyID, WellName	WellSiteID
ldgTank	4.1 - 4.2	✓		stamp	TankID, RefDocID, RefDocType, ProductID
lstActionTaken	4.2	□		n/a	ActionTakenID
lstAutoDeviceManufacturer	4.1 - 4.2			n/a	AutoDeviceManufacturerID
lstAutomationDeviceType	4.1 - 4.2			n/a	AutomationDeviceTypeID
lstChemAppMethod	4.1 - 4.2			n/a	ChemAppMethodCode
lstChemFlushAgent	4.1 - 4.2			n/a	ChemFlushAgentCode
lstChemicalCompany	4.1 - 4.2			n/a	ChemCoCode
lstChemicalType	4.1 - 4.2			n/a	ChemTypeCode
lstCompFailure	4.1 - 4.2			n/a	CompressorFailID

lstDocSource	4.1 - 4.2			n/a	DocSourceID
lstDocType	4.1 - 4.2			n/a	DocTypeID
lstDownTime	4.1 - 4.2			n/a	DownTimeID
lstECDDownTime	4.2			n/a	ECDDownTimeID
lstECDEntityType	4.2			n/a	ECDEntityTypeID
lstElecMotorFailure	4.1 - 4.2			n/a	ElecMotorFailID
lstElecSubPumpFailure	4.1 - 4.2			n/a	ElecSubPumpFailID
lstEngineFailure	4.1 - 4.2			n/a	EngineFailID
lstEntityType	4.1 - 4.2			n/a	EntityTypeID
lstEquipmentType	4.1 - 4.2			n/a	EquipTypeID
lstfdStatus	4.1 - 4.2			n/a	StatusID
lstFuelType	4.1 - 4.2			n/a	FuelTypeID
lstGasDispReason	4.1 - 4.2			n/a	GasDispID
lstInjectionPointType	4.1 - 4.2			n/a	InjectionPointTypeID
lstLiquidXFerReason	4.1 - 4.2			n/a	LiquidXferID
lstMetal	4.1 - 4.2			n/a	MetalID
lstMeterType	4.1 - 4.2			n/a	GasMeterTypeID
lstOilDispReason	4.1 - 4.2			n/a	OilDispID
lstPlungerType	4.1 - 4.2			n/a	PlungerTypeID
lstProducingMethod	4.1 - 4.2			n/a	ProducingMethodID
lstProducts	4.1 - 4.2			n/a	ProductID
lstPumpFailure	4.1 - 4.2			n/a	PumpFailID
lstPumpingUnitModel	4.1 - 4.2			n/a	PumpingUnitModelCode
lstPumpUnitFailure	4.1 - 4.2			n/a	PumpUnitFailID
lstRodFailure	4.1 -			n/a	RodFailID

	4.2				
lstRodPumpFailure	4.1 - 4.2			n/a	RodPumpFailID
lstRodPumpType	4.1 - 4.2			n/a	RodPumpTypeCode
lstStatus	4.1 - 4.2			n/a	StatusID
lstTankType	4.1 - 4.2			n/a	TankTypeID
lstTubingFailure	4.1 - 4.2			n/a	TubingFailID
lstUnits	4.1 - 4.2			n/a	UnitsCode
lstUserPropAccessType	4.1 - 4.2			n/a	UserAccessTypeID
lstUserType	4.1 - 4.2			n/a	UserTypeID
lstValveType	4.1 - 4.2			n/a	ValveTypeID
lstWaterDispReason	4.1 - 4.2			n/a	WaterDispID
lstWellTestMethod	4.1 - 4.2			n/a	WellTestMethodID
ptsChemicalInjPt	4.1 - 4.2	✓		stamp	ChemicalInjPtID
ptsInjectionPt	4.1 - 4.2	✓		PropertyID, WellSiteID, Description	InjectionPtID
ptsMeterPt	4.1 - 4.2	✓		PropertyID, GatheringSiteID, Description	MeterPtID
ptsProductionPt	4.1 - 4.2	✓		PropertyID, Description	ProductionPtID
sumDailyInjection	4.1 - 4.2	✓	✓	stamp	EntityID, EntityType, ProductID, docDate
sumDailyProduction	4.1 - 4.2	✓	✓	stamp	EntityID, EntityType, docDate

FDDS Beta Notes

1. Future versions of FDDS will return header information which will enable you to determine if all records have been received. Until then, you can assume that if fewer than 100,000 records are included in the dataset, all records have been received.
2. All required FDDS URL parameters must be provided, even if they are not used in the subsequent query. (The URL syntax presented in *section 'The FDDS URL' page 11* encloses optional parameters inside square brackets). For example, when retrieving the "lst" tables, the stampstartdate and projectstartdate parameters are not used in the FDDS query. However, FDDS will throw an error if these parameters are not provided.

FAQ s received from our customers

For some tables, FDDS returns more records than a Production Explorer update returns. Why?
Some tables have a status field which is set to "deleted". A PE update does not return these records, but FDDS does. Example: All of the equipment tables (eqpTank, eqpMeter, etc.) have an fdStatus field which is populated with one of the values listed in the lstfdStatus table. If the field is set to lstfdStatus.Deleted, the corresponding record will not be downloaded during a PE update. It will, however, be downloaded using FDDS.

For some tables, it doesn't appear that the stamp start date is working correctly. We're supplying a stamp start date, but the web service is pulling down all records in that table. What's going on?
You might notice this behavior for the eqpxxx tables or the ptsxxx tables. Remember that the service is returning only those records which are connected with properties that are assigned to the user specified by the PARM_USERID parameter. The query therefore involves another table,

fd_sysUserProperties. You can think of a ptsxxx record or an eqpxxx record as consisting of both the physical equipment information and the relevant assignments. If either the physical record stamp date or the corresponding assignment record stamp date is greater than the stamp start date, the record is included. Production Explorer downloads use the same logic. In fact, our PE customers requested this behavior. For example, let's say the last time a production point was updated was on 09/1/2011, and one or more users were assigned to that production point on 11/3/2011. If one of these assigned users uses FDDS or PE to download the production point table and specifies a stamp start date of 11/3/2011, he or she would expect to see some production point records.

I have an FDDS job which runs several times throughout the day to download docxxx table records. When a job runs, I save the last stamp date returned from each downloaded doc table and use it as my stamp start date parameter. Am I guaranteed to receive all records without duplicates using this method? If not, what should I do?

The short answer is that you are guaranteed to receive all records using this approach. You may or may not receive duplicate records. In our opinion, though, this is the best retrieval method. If you use a stamp start date which is greater than the last record received, you may miss some records. Your retrieval program should therefore include some logic to weed out the duplicate records. One client designed their system to first check to see if a record with the primary key value already existed in their local database. If so, their program performs an update instead of an insert. A list of all primary keys is included in *Section 'Using the FDDS URL – some examples' page 15* of this document.

Some background… The stamp field in our doc tables is stored as a SQL Server™ datetime field. It is therefore accurate down to one three-hundredth of a second (equivalent to 3.33 milliseconds or 0.00333 seconds. In addition, SQL Server™ rounds the stored values to increments of .000, .003, and .007. Examples: 12/12/2011 03:12:55.557 AM, 12/12/2011 03:12:55.600 AM. It is not uncommon for different doc records to have the same stamp date. Some of our customers use custom data loaders which perform bulk data uploads. In this case, many records may have exactly the same stamp date, down to the rounded millisecond. If you pull data using FDDS and catch the database in the middle of such a load, there will be new records with exactly the same stamp date the next time you pull data. If you started with a stamp start date that was greater than the last stamp date received, you would miss these new records.

Customer Care Contact Information

If you have any inquires or need any assistance please contact your local Customer Care Representative.

US Telephone: +1 800 447 2273

Email: Customercare@ihs.com

If you have any comments or suggestions for new features or improvements to FieldDIRECT Data Services (FDDS) please do not hesitate to contact us.