**IDQ Parser Transformation**

It is one of most important transformation used in IDQ. Parsing is the core function of any data quality tool and IDQ provides rich parsing functionality to handle complex patterns.  
  
Parser transformation can be created in two mode

* Token Parsing Mode
* Pattern Based Parsing

**Token Based Parsing** : It is used to parse strings that match token sets regular expression or reference table based entries.We will use a simple example to create a token based parser transformation.Suppose we have email id coming in a field in format "Name@company.domain" and we want to parse this and store it in multiple fields

NAME COMPANY\_NAME DOMAIN

Suppose we have input data coming as below

Rahul@gmail.com

Sachin@yahoo.com

Stuart@yahoo.co.uk

We will create a token based parser transformation having email id as input ,After creating transformation go to properties and strategies tab and click on new

Token Based Parsing :It is used to parse strings that match token sets regular expression or reference table based enteries.

We will use a simple example to create a token based parser transformation.Suppose we have email id coming in a field in format "Name@company.domain" and we want to parse this and store it in multiple fields

NAME

COMPANY\_NAME

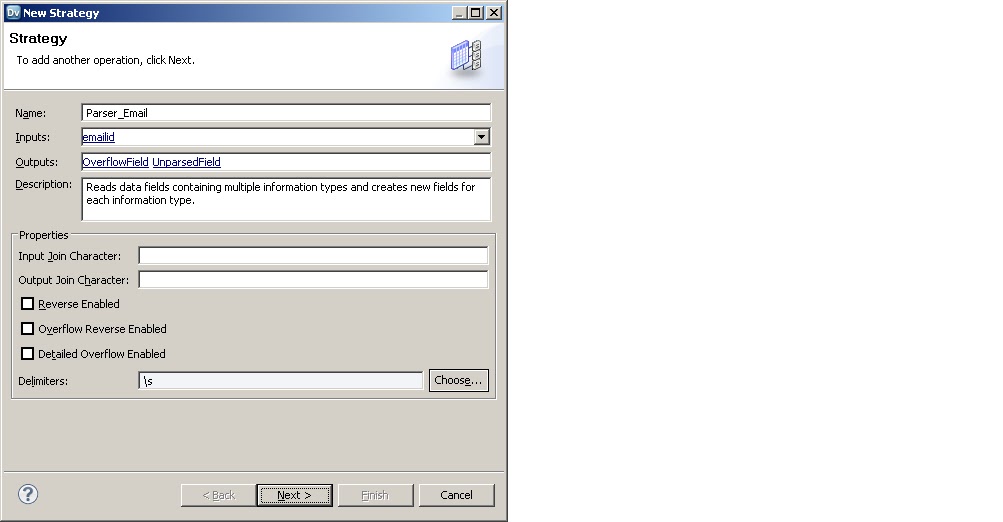
DOMAIN

Suppose we have input data coming as below

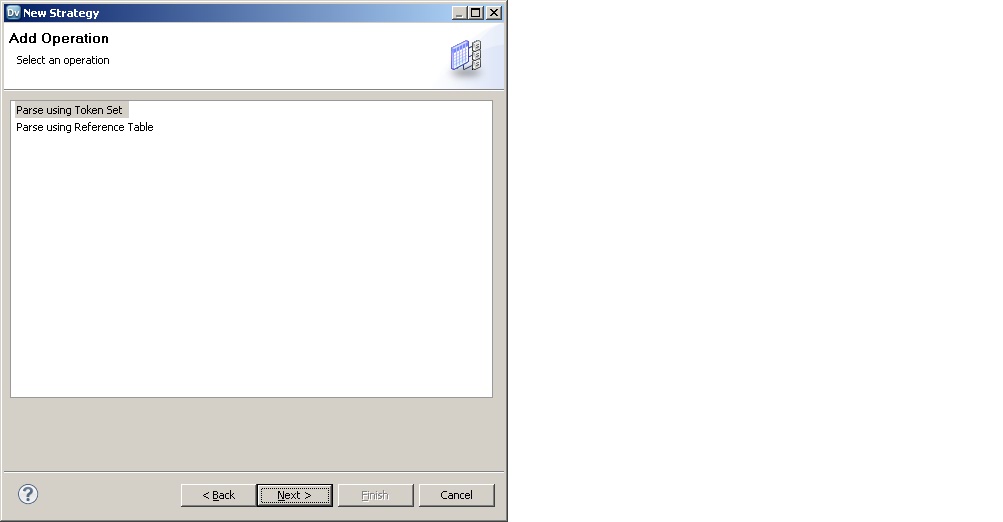
Rahul@gmail.com

Sachin@yahoo.com

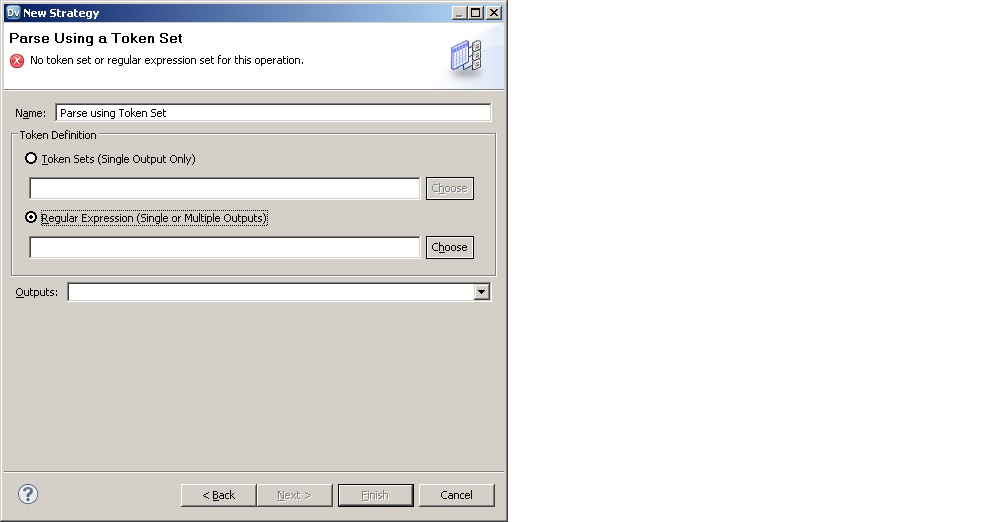
Stuart@yahoo.co.uk  
  
**Step1 :**We will create a token based transformation having email id as input ,After creating transformation go to properties and strategies tab and click on new

[](http://3.bp.blogspot.com/-CWFO23iqGp0/UnA_TRp2HjI/AAAAAAAAAq0/UlYKThxVxgo/s1600/Parser_email1.bmp)

**Step2 :** Click on Token Based

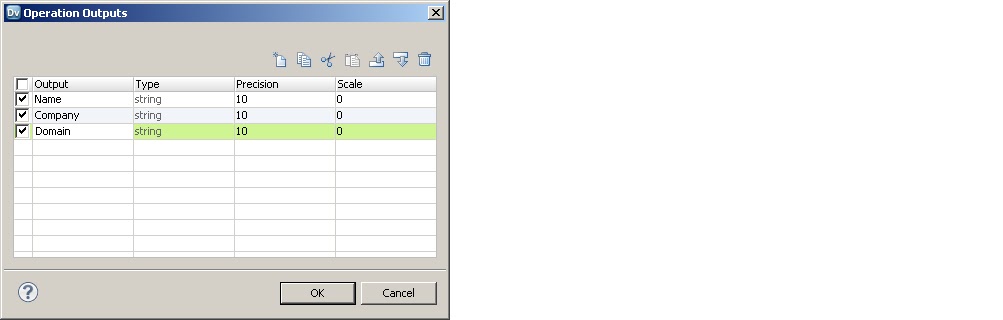
[](http://2.bp.blogspot.com/-j4Y8NT7qY8A/UnA_9VqAKVI/AAAAAAAAAq8/OAg9CxdUFLk/s1600/parser_2.bmp)

**Step3 :** Select Regular expression (As we want to have multiple output port)

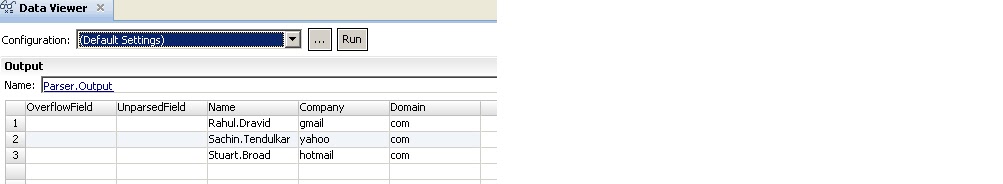
[](http://4.bp.blogspot.com/-BeZ-zdaB8no/UnBAZ0y16zI/AAAAAAAAArE/MUpzrt-klIY/s1600/parser3.bmp)

**Step4)** Select email parser or you can create your own regular expression to parse different type of transformation

**Step5)** Create three output port and click on OK then finish

[](http://1.bp.blogspot.com/-HrsWNNtTkF0/UnBBv426qsI/AAAAAAAAArY/nvCMh_r6f-A/s1600/parser6.bmp)

Below is output from Parser transformation Name ,company and email id parsed into separate fields.

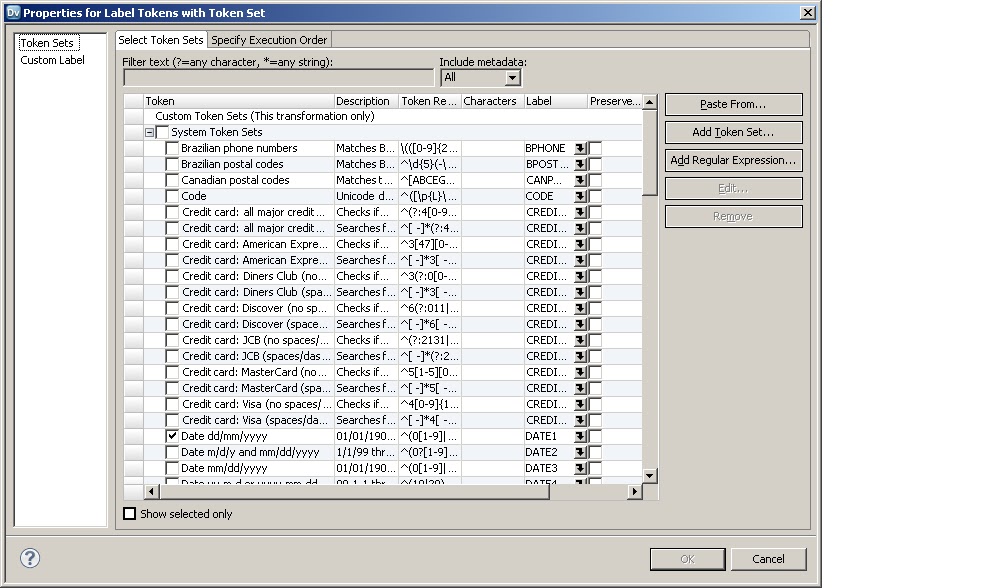
[](http://1.bp.blogspot.com/-axXzguOH3JA/UnBAwNVaGcI/AAAAAAAAArQ/0M3bwQ7LFVU/s1600/parser4.bmp)

**Pattern Based Parsing** : Pattern based parsers are useful when working with data that needs to be parsed apart or sorted and the data has a moderately high number of patterns that are easily recognized.

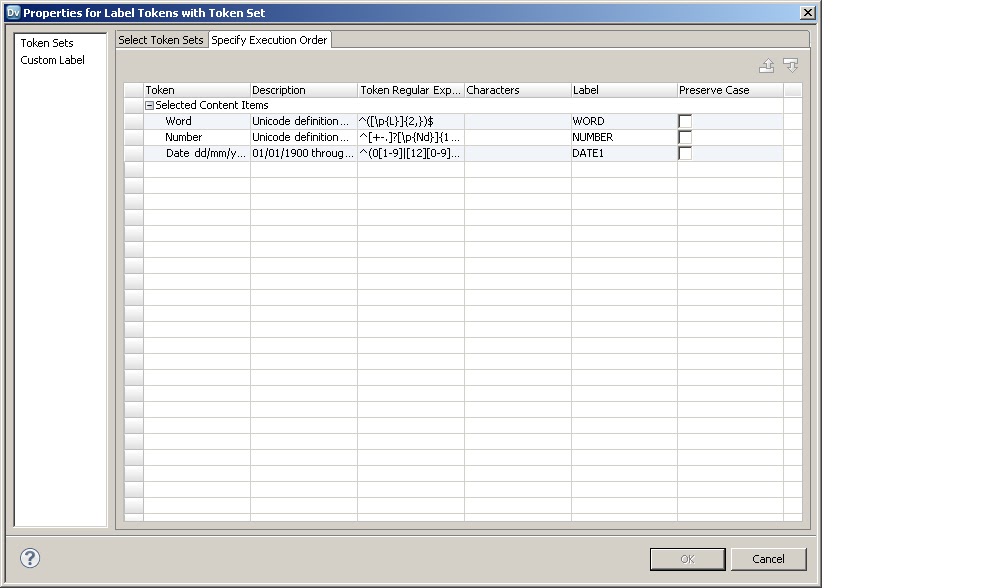
Parser Based Transformation need to have output from Label Transformation which will provide two outputs LabelData and Tokenised data

Suppose we have a field named as PATTERN\_DATA in source which contains name ,empno and date in it and we need to parse into three seperate fields

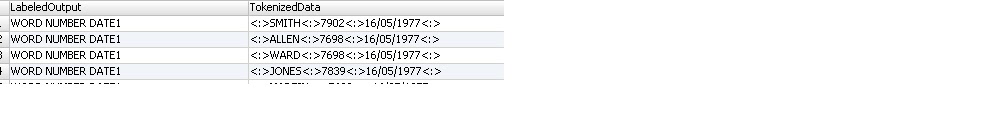
**Step1 )** We will first create a label transformation with delimiter as , and below properties by creating new strategies

[](http://1.bp.blogspot.com/-rq5uG1wMEKo/UnBC-m5lL-I/AAAAAAAAArk/jR_cee3s1TI/s1600/Parser7.bmp)

in second tab choose execution order and assign label

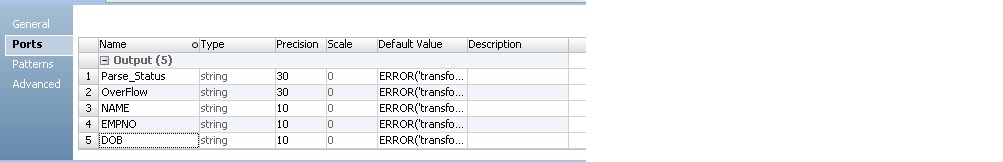
[](http://4.bp.blogspot.com/-pAGXXObZeD4/UnBDWI11F4I/AAAAAAAAArs/HjX_Xu0iABY/s1600/parser8.bmp)

    Output of Label transformation will be

[](http://3.bp.blogspot.com/-edGskDqeWYI/UnBDpGB4cKI/AAAAAAAAAr0/jFGHBgDhjtw/s1600/parser9.bmp)

**Step2 )** Connect both LabeledOutput and Tokenized data to pattern based transformation

and create three new output port in port tab as shown below

[](http://4.bp.blogspot.com/-KpMN1_HlSGw/UnBEV_YcbNI/AAAAAAAAAr8/s2JPcrTNUzQ/s1600/parser10.bmp)

**Step3 )** In Pattern Tab define below (As per Label defined in Label)

[](http://3.bp.blogspot.com/-Kn8t2ltuQhc/UnBEtJQerjI/AAAAAAAAAsE/Eq93WH_L8zE/s1600/parser11.bmp)

You can preview Parser data broken in three fields NAME EMPNO DOB

========================================================================

Q1. EXPLAIN THE IDQ FUNCTIONALITY?

Use the IDQ to design and run processes to complete the following tasks:

Profile data : Profiling reveals the content and structure of data. Profiling is a key step in any data project, as it can identify strengths and weaknesses in data and help you define a project plan.

Create scorecards to review data quality : A scorecard is a graphical representation of the quality measurements in a profile.

Standardize data values : Standardize data to remove errors and inconsistencies that you find when you run a profile. You can standardize variations in punctuation, formatting, and spelling. For example, you can ensure that the city, state, and ZIP code values are consistent.

Parse data : Parsing reads a field composed of multiple values and creates a field for each value according to the type of information it contains. Parsing can also add information to records. For example, you can define a parsing operation to add units of measurement to product data.

Validate postal addresses : Address validation evaluates and enhances the accuracy and deliverability of postal address data. Address validation corrects errors in addresses and completes partial addresses by comparing address records against address reference data from national postal carriers. Address validation can also add postal information that speeds mail delivery and reduces mail costs.

Find duplicate records : Duplicate analysis calculates the degrees of similarity between records by comparing data from one or more fields in each record. You select the fields to be analyzed, and you select the comparison strategies to apply to the data. The Developer tool enables two types of duplicate analysis: field matching, which identifies similar or duplicate records, and identity matching, which identifies similar or duplicate identities in record data.

Create reference data tables : Informatica provides reference data that can enhance several types of data quality process, including standardization and parsing. You can create reference tables using data from profile results.

Create and run data quality rules : Informatica provides rules that you can run or edit to meet your project objectives. You can create mapplets and validate them as rules in the Developer tool.

Collaborate with Informatica users :The Model repository stores reference data and rules, and this repository is available to users of the Developer tool and Analyst tool. Users can collaborate on projects, and different users can take ownership of objects at different stages of a project.

Export mappings to PowerCenter : You can export mappings to PowerCenter to reuse the metadata for physical data integration or to create web services.

Q1 What is the difference between the Power Center Integration Service and the Data Integration Service?

The Power Center Integration Service is an application service that runs sessions and workflows.

The Data Integration Service is an application service that performs data integration tasks for the Analyst tool,the Developer tool, and external clients.

The Analyst tool and the Developer tool send data integration task requests to the Data Integration Service to preview or run data profiles, SQL data services, and mappings.

Commands from the command line or an external client send data integration task requests to the Data Integration Service to run SQL data services or web services.

Q2.What is the difference between the PowerCenter Repository Service and the Model Repository Service?

The PowerCenter application services and PowerCenter application clients use the PowerCenter Repository Service. The PowerCenter repository has folder-based security.

The other application services, such as the Data Integration Service, Analyst Service, Developer tool, and Analyst tool, use the Model Repository Service.

The Model Repository Service has project-based security.

You can migrate some Model repository objects to the PowerCenter repository.

Q3 What is the difference between a mapplet in PowerCenter and a mapplet in the Developer tool?

Mapplet in PowerCenter and in the Developer tool is a reusable object that contains a set of transformations. You can reuse the transformation logic in multiple mappings.

PowerCenter mapplet can contain source definitions or Input transformations as the mapplet input. It must contain Output transformations as the mapplet output.

Developer tool mapplet can contain data objects or Input transformations as the mapplet input. It can contain data objects or Output transformations as the mapplet output.

A mapping in the Developer tool also includes the following features:

You can validate a mapplet as a rule.

You use a rule in a profile.

A mapplet can contain other mapplets.

Q4 What is the difference between a mapplet and a rule?

You can validate a mapplet as a rule. A rule is business logic that defines conditions applied to source data when you run a profile.

You can validate a mapplet as a rule when the mapplet meets the following requirements:

It contains an Input and Output transformation.

The mapplet does not contain active transformations.

It does not specify cardinality between input groups.

Q5 What is the difference between a source and target in PowerCenter and a physical data object in the Developer tool?

In PowerCenter, you create a source definition to include as a mapping source. You create a target definition to include as a mapping target. In the Developer tool,

you create a physical data object that you can use as a mapping source or target.

Q6 What is the difference between the PowerCenter Repository Service and the Model Repository Service?

The PowerCenter application services and PowerCenter application clients use the PowerCenter Repository Service. The PowerCenter repository has folder-based security.

The other application services, such as the Data Integration Service, Analyst Service, Developer tool, and Analyst tool, use the Model Repository Service.

The Model Repository Service has project-based security.

You can migrate some Model repository objects to the PowerCenter repository.