

ITP

ITP/MDK Repository

2003-01-24

*This whitepaper described the basic functionality of the ITP/MDK
Repository*



INTELLIGENT TEXT PROCESSING

Table of contents

Table of contents	1
1. Introduction	2
2. Introduction to ITP	3
2.1 Model documents, models, result documents and DID files.....	3
2.2 The ITP document development process	3
2.3 Producing result documents	4
3. Features of the ITP/MDK repository	5
4. Technical background	6
4.1 The ITP/MDK Repository database	6
4.1.1 Supported databases.....	6
5. Interfaces	7
5.2 The ITP/MDK development environment.....	7
5.3 Two different views.....	8
6. Using the ITP/MDK Repository	9
6.2 Revisions	10
6.3 Labels	11
6.4 Central storage of include documents	11
6.5 Deploy	12
7. Authorisation.....	13

1. Introduction

This whitepaper describes the basic functionality of the ITP/MDK Repository product. ITP (model) documents play an important role in our customers' business. We found that managing these documents becomes more and more important. Several customers started using generic revision control systems to manage these documents. That's when we decided on developing a Document Management System, especially targeting maintenance of ITP model documents and ITP models. This has become the ITP/MDK Repository.

2. Introduction to ITP

ITP model documents, the template documents, are developed in the word processor of your choice. After development of a model document, ITP creates the final output document(s) based on this model document. The word processor itself is not used during this creation process.

ITP combines all familiar word processor facilities, both textual as well as layout, in a seamless way with data from your corporate databases. Text and layout in the documents can be conditioned based on the data from the database. ITP accesses the database in real-time, so no downloading, pre-processing or copying of data is required.

Document production can be document driven, where the output is based on conditions and calculations coded in your document that may or may not use database access. It can also be data driven, where the output is based on the contents of the database. If necessary, the document production or parts of it can be controlled by user interaction.

2.1 Model documents, models, result documents and DID files

ITP works with four kinds of files in the text/data merge process:

- **Model documents:** These are word processor documents that can contain all kinds of text with all layout facilities offered by the word processor, mixed with ITP instructions. The ITP instructions enable data retrieval using external retrieval programs or logic, manipulation of data and merging those data into the document text. A model document is the equivalent of a program source.
- **DID files:** The Database Interface Definition describes the structure of the data that is used. The DID describes how to access the database using for example SQL queries or external data retrieval programs. DID files are only used during the generation of ITP Models from ITP model documents.
- **ITP Models:** These are binary files generated from model documents, containing an optimized form of the model document, including all text, layout and ITP instructions. Model documents are compiled to ITP models. The model is the equivalent of a program.
- **Result documents:** These are normal word processor documents that contain the final document text, merged with -possibly manipulated- data. These documents are produced by executing ITP models.

2.2 The ITP document development process

Developing a document in ITP means writing model documents (in your favorite word processor), compiling them to ITP models and testing them.

ITP program logic is written in the model document using the ITP instruction language, a powerful yet easy to use language, which has two very special properties, concerning input/output:

- Input is always done through a highly abstract and structured interface to the database or to external programs (DID files). The actual database access is not programmed in the

model itself. Furthermore some user interaction can take place for additional input. All other input for the result document is already present in the model document itself! Other means of input are not (and do not have to be) available.

- Output is the result document itself, which means that all output produced is text and layout for the result document(s). Both may of course depend on database content. No special output facilities are needed.

The ITP instruction language furthermore offers all the usual programming facilities: functions, if-then clauses, while-loops, variables, arrays etc. It also offers powerful calculation, data and text manipulation and conversion functions. A model can be designed to produce one or more result documents.

The ITP instruction language is in fact a kind of programming language on word processor document level. Contrary to a normal programming language it is very easy to use (even by trained end-users), because of the special ways in which is dealt with input and output and because of the focus on text and layout in the word processor's native environment.

2.3 Producing result documents

Executing the corresponding ITP models generates result documents; this is called the RUNMDL process. This process can be initiated from the ITP user interface and the ITP development environments or it can be integrated in an application, using ITP's runmdl API.

While running the model users can be asked to make (database) selections or to provide additional input. This is the case with interactive models. Models can also be designed to run without any intervention by the user.

During one run of the model, one or more result documents can be produced.

The runmdl process is a highly optimized process. Performance is very good; limited mainly by how fast data can be retrieved.

The result document is in the same word processor format as the original model document.

3. Features of the ITP/MDK repository

Overview of the most important features of the ITP/MDK Repository:

- Central storage of documents, models and DID files, supporting back up and security.
- Project Organisation of documents, models and DID files.
- Revision control.
- Reporting facilities.
- User activity tracking.
- Multi-User Access control, protecting document integrity.
- Dependency tracking.
- Document Editing Facilities.
- ITP Development Support for compiling and running models.

4. Technical background

The ITP/MDK Repository consists of a client component and a server component. The ITP/MDK Repository client reports to the ITP/MDK Repository server during login. The ITP/MDK Repository server checks the licence and grants access to the ITP/MDK Repository server database.

4.1 The ITP/MDK Repository database

All ITP/MDK Repository objects are stored in a (central) database. This provides security and supports back-ups.

4.1.1 Supported databases

In principle, the ITP/MDK Repository supports most ODBC enabled databases. However because of the advanced ODBC functions used by the ITP/MDK Repository, using the ITP/MDK Repository with databases that have not been verified for the ITP/MDK Repository is not supported. We tested the with following databases:

Microsoft SQL Server (version 7 and up)

Tested with MS ODBC driver 2000.81.9001.00.

Oracle (version 8i and up)

Tested with ODBC driver 8.01.77.00.

Microsoft Access

Tested with ODBC driver Access 4.00.5303.01. *Aia does not support using Access for anything else than demonstration purposes.*

DB2 UDB for iSeries

Tested with Client Access version 8.00.04.08, OS/400 version V4R5 with all available PTFs installed.

The ITP/MDK Repository uses transactions to perform modifications to the database. Therefore, the library in which the ITP/MDK Repository database will be stored must be created with a journal.

MySQL

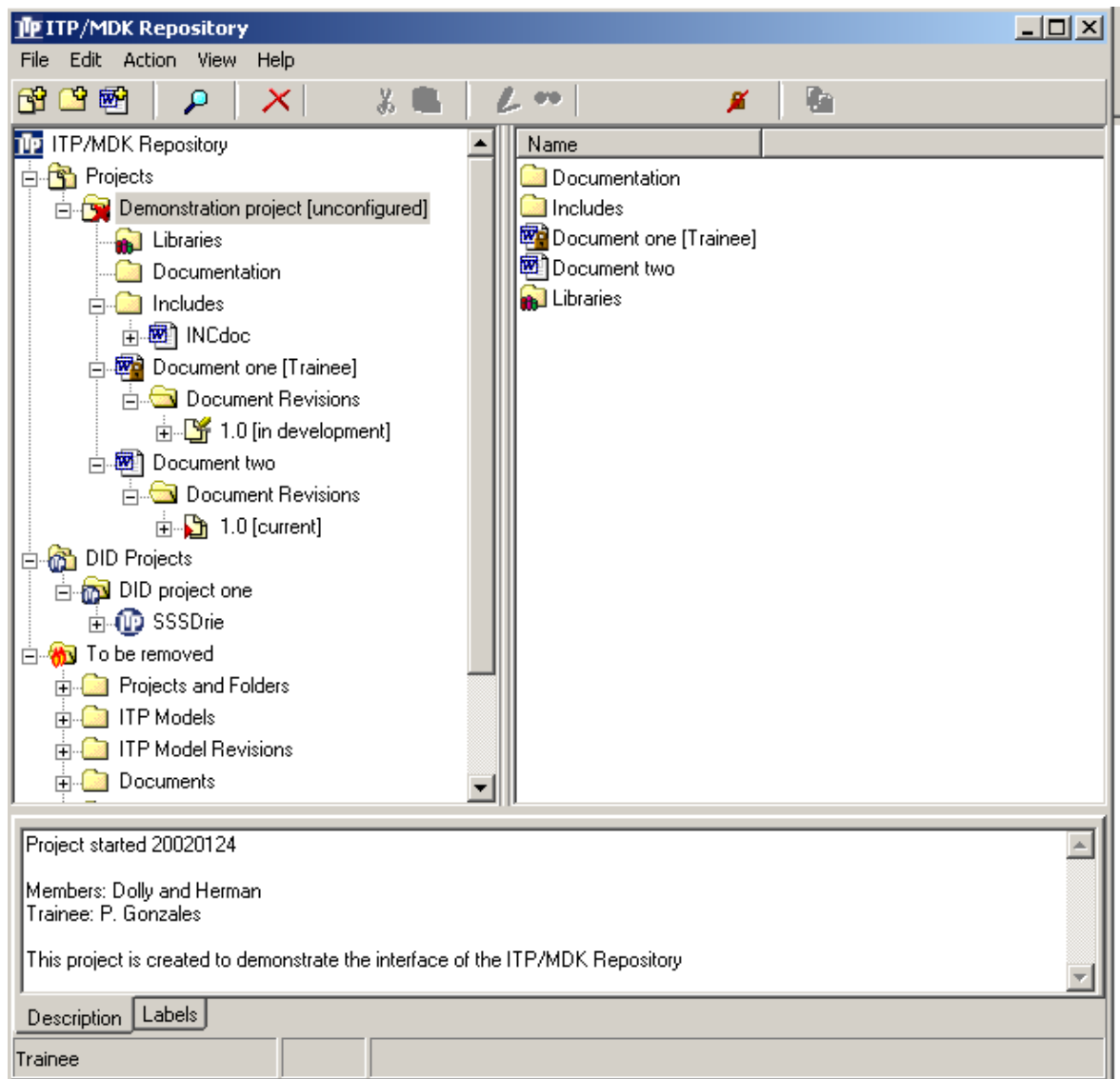
Tested with ODBC driver SQL Server 3.70.08.21.

We have tested the ITP/MDK Repository with version 2.5 of the MySQL software. This version is recommended. Later versions were not stable at the moment this document was written.

5. Interfaces

The main window of the ITP/MDK Repository is divided in several panes through which all functionality of the ITP/MDK Repository can be reached.

The ITP/MDK Repository main window



Whitepaper

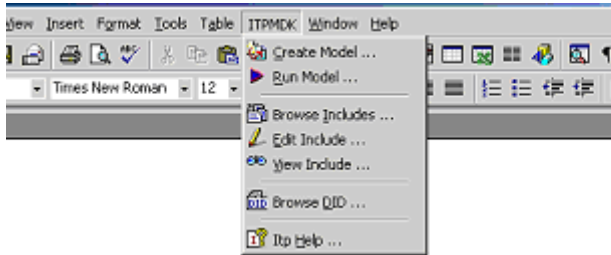
5.2 The ITP/MDK development environment

The ITP model documents and ITP include documents can be opened for viewing or editing from within the ITP/MDK Repository. The documents will be opened in the ITP/MDK Repository's Word based development environment. A special ITP menu, ITP toolbar and styles toolbar are added to support the development of ITP model documents and ITP include documents.

ITP – ITP/MDK Repository

The ITP/MDK Word Menu

From within the ITP menu, it is possible to perform the most important tasks involved in developing ITP model documents such as creating and running models.



#BEGIN

END#

The ITP/MDK Word Toolbar

The ITP/MDK toolbar offers support for the layout of model document. Styles are available to improve readability of ITP model documents and ITP include documents.



DID files

DID files can be imported in the ITP/MDK Repository from the file system providing revision control.

5.3 Two different views

Different users have to perform different tasks. Not all information in the ITP/MDK Repository is of interest to all users. The ITP/MDK Repository offers two main views on its database: one view in which all versions of a document are shown and another view in which only the active version of the documents is shown. The latter view will be most appropriate for standard document development tasks.

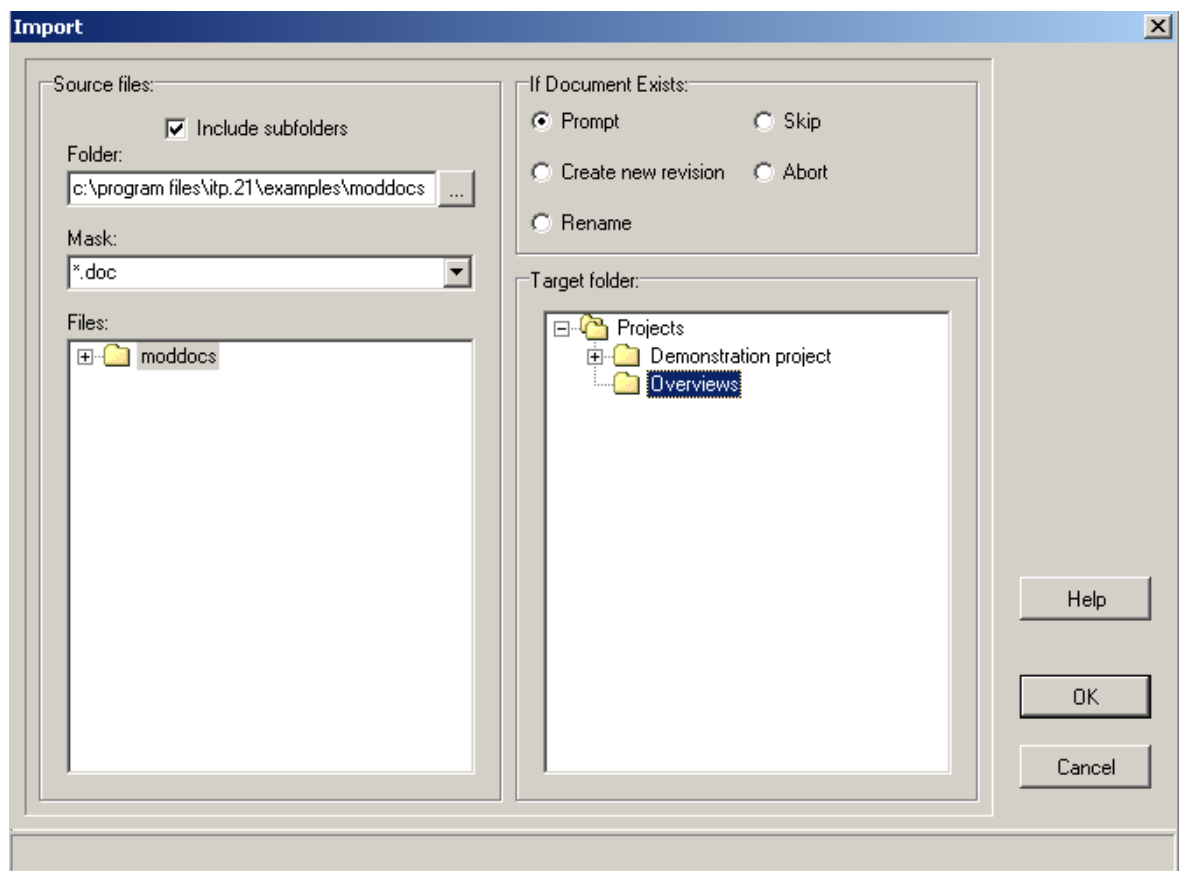
6. Using the ITP/MDK Repository

Roughly, new users of the ITP/MDK Repository can be divided into two groups:

1. Customers who already use ITP
2. Customers who are new to ITP

Customers who already use ITP probably already developed a large number of ITP model documents and include documents. The ITP/MDK Repository offers functions to import these documents.

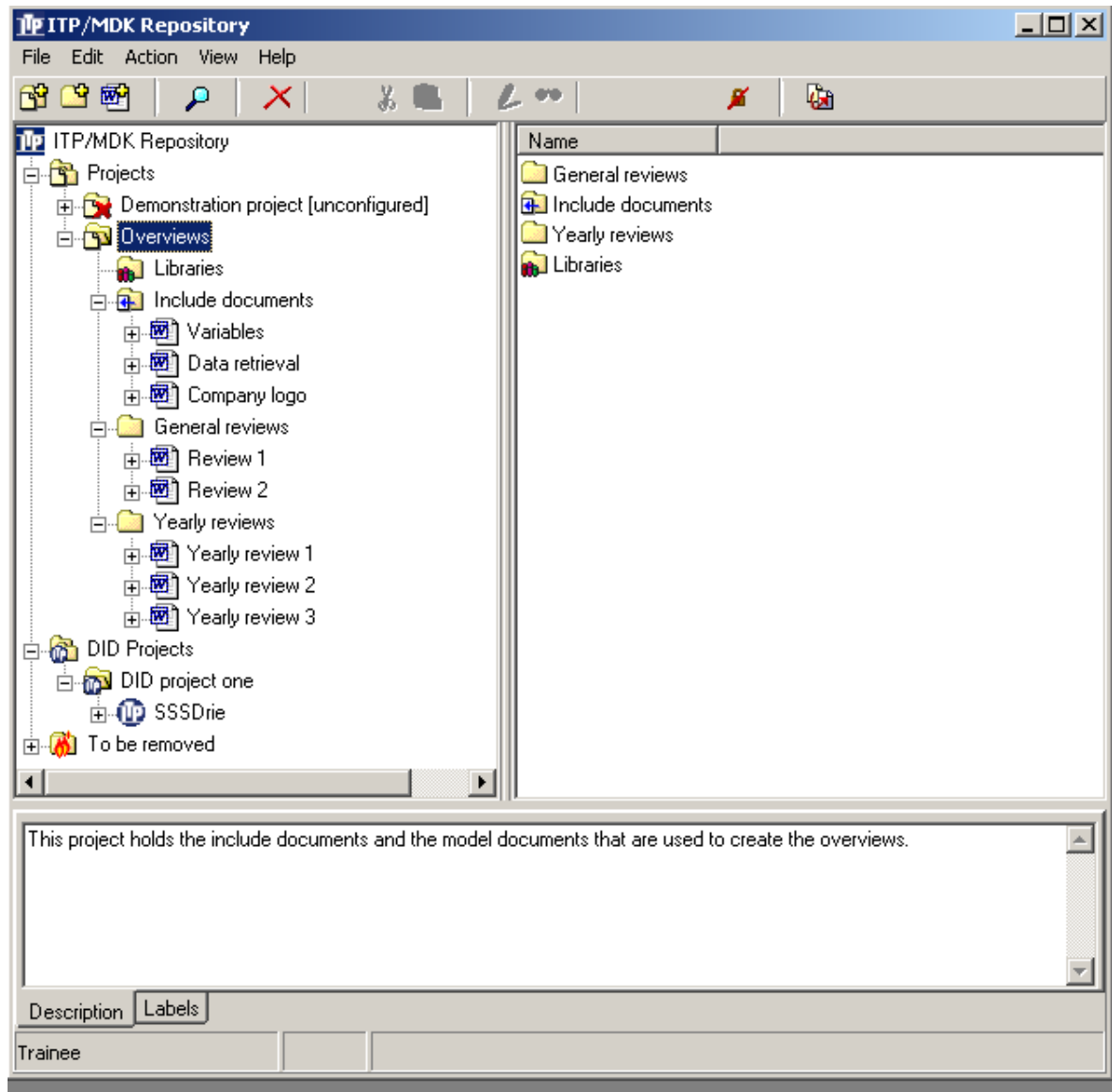
The import window



ITP – ITP/MDK Repository

Customers who are new to ITP will set up a project organisation in the ITP/MDK Repository from which they start developing their documents.

Example project set-up



Documents can be opened to view them or to edit them in the ITP/MDK. The ITP/MDK offers a Word based environment with support for the development of ITP model documents and include documents.

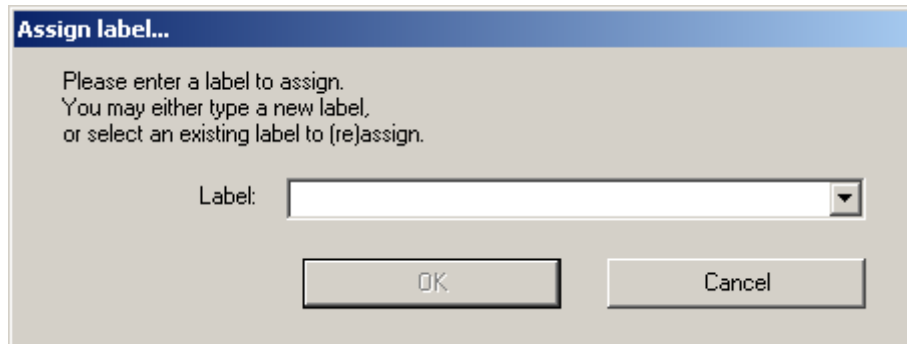
6.2 Revisions

When a document is opened for editing by a user in the ITP/MDK, that user locks that document. This means that other users are not allowed to edit that document. This guarantees the consistency of the ITP/MDK Repository database.

A new revision of a document is created at the moment that the user who edits the document unlocks the document. This new version will automatically become the current revision. Older revisions of a document can at any moment be set as the current revision.

6.3 Labels

Revisions in the ITP/MDK Repository can be labelled with an unlimited number of labels per revision. This can be used to keep track why a certain revision was created and which revisions were deployed at a certain moment.



6.4 Central storage of include documents

The use of include documents is essential for good model development practices. The ITP/MDK Repository supports the use of include documents. Include paths can be set at several levels in the ITP/MDK Repository. Every project can contain one or more include folders. Besides these, special library projects can be created. These libraries can be included in multiple projects in the ITP/MDK Repository thus allowing these projects to use the documents in its include paths(s).

6.5 Deploy

At a certain time in the development process, the ITP model(s) will have to be exported to the file system. With the Deploy function all ITP models together with the appropriate configuration files and DID files are exported to the file system. These models can then be used without any changes.

The Deploy window

Deploy...

Please select a target directory for the Models, and indicate which revisions should be deployed and which label they should get.

Select directory

c:\itpwork - Overviews Select directory

Deploy method

Current revision

Assign label

Deploy 2003.01.22 16:48:20

Labelled revision

OK Cancel


7. Authorisation

Different users have different tasks. For example some of them will be concerned with the project organisation, others will be limited to the maintenance of existing documents. The ITP/MDK Repository will support these different roles in its authorisation options. A role is granted access to a set of actions, like “edit”, “create folder”, and so on.

Furthermore, the ITP/MDK Repository will offer authorisation on projects. Users can be granted access to projects or not.

These two authorisation options can be combined thus offering a powerful way of controlling the use of the ITP/MDK Repository.

Authorisation will be added in a future version of the ITP/MDK Repository.

ITP is developed by 

For more information please contact us.

Telephone: +31 24 371 02 30
Fax: +31 24 371 02 31
WWW: <http://www.aia-ity.com/>
Email: info@aia-ity.com
Postal Address: P.O. Box 38090
6503 AB Nijmegen
The Netherlands
Visiting Address: Kerkenbos 10-129
6546 BJ Nijmegen
The Netherlands