IEC 62769-6-100

Edition 1.0 2023-04

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

Field Device Integration (FDI®) – Part 6-100: Technology Mapping – .Net

Intégration des appareils de terrain (FDI®) – Partie 6-100: Mapping de technologies – .Net

IEC 62769-6-100:2023-04(en-fr)

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC
La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC
Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC - webstore.iec.ch/advsearchform
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, …). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IEC 62769-6-100

Edition 1.0 2023-04

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour
inside

Field Device Integration (FDI®) –
Part 6-100: Technology Mapping – .Net

Intégration des appareils de terrain (FDI®) –
Partie 6-100: Mapping de technologies – .Net

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-6809-4

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

FIELD DEVICE INTEGRATION (FDI®) –

Part 6-100: Technology Mapping – .NET

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 62769-6-100 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65E/868/CDV	65E/925/RVC

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

A list of all parts in the IEC 62769 series, published under the general title *Field device integration (FDI®)*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

FIELD DEVICE INTEGRATION (FDI®) –

Part 6-100: Technology Mapping – .NET

1 Scope

This part of IEC 62769 specifies the technology mapping for the concepts described in the Field Device Integration (FDI®[1]) standard. The technology mapping focuses on implementation regarding the components FDI® Client and User Interface Plug-in (UIP) using the Runtime .NET. This runtime is specific only to the WORKSTATION platform as defined in IEC 62769-4.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62769-1:2021, *Field device integration (FDI®) – Part 1: Overview*

IEC 62769-2, *Field device integration (FDI®) – Part 2: Client*

IEC 62769-4, *Field device integration (FDI®) – Part 4: FDI® Packages*

IEC 62769-6, *Field device integration (FDI®) – Part 6: Technology Mappings*

3 Terms, definitions, abbreviated terms, acronyms and symbols

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1 as well as the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http:www.iso.org/obp

3.1.1
Application Domain
isolated environment where applications execute

3.1.2
Assembly
reusable, version information providing, and self-describing building block of a CLR application

[1] FDI is a registered trademark of the non-profit organization Fieldbus Foundation, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trade name. Use of the trade name requires permission of the trade name holder.

3.1.3
FDI® Type Library
typescript file that contains the interfaces and data types that are used for the data exchange and interaction between a UIP and an FDI® Client

3.1.4
Global Assembly Cache
machine-wide code cache that stores Assemblies specifically designated to be shared by several applications

3.1.5
Windows Registry
system-defined database in which applications and system components store and retrieve configuration data

3.2 Abbreviated terms and acronyms

For the purposes of this document, the abbreviated terms and acronyms given in IEC 62769-1, IEC 62769-6, as well as the following apply.

MSI	Microsoft Installer
WPF	Windows Presentation Foundation
UML	Unified Modeling Language

3.3 Symbols

Figures in this document use the graphical symbols according to l'ISO/IEC 19505-1 (UML 2.0).

4 Technical concepts

4.1 General

4.1.1 Overview

In 4.1.2, 4.2, 4.3, 4.4, and 4.5, this document describes the technology base for UIP implementation based on the runtime .NET Framework CLR4, the hardware and software environment including the related implementation rules. Clause 4 follows a lifecycle (use case) oriented approach.

Subclause 4.6 describes the copy deployment procedures and related implementation rules for the UIP and the FDI® Client. UIP executable instantiation and termination is described in 4.7. Subclause 4.8 defines the rules about interaction between the FDI® Client and the UIP. Security related definitions are written in 4.9. The service interface definitions for the FDI® Client and the UIP are found in Clause 5.

4.1.2 FDI® Type Library

The Device Access Services and the UIP Services can be modelled as .NET interfaces passing .NET data type arguments. These interfaces and data types are used for the data exchange and interaction between the UIP and the FDI® Client. For runtime error handling purposes during interface method calls .NET exceptions classes are defined.

The FDI® .NET interfaces, data types, and exception classes are defined in a single FDI® Type Library. The FDI® Type Library is provided within a Nuget Package, which contains one or more strong named assemblies. The file name of this Nuget Package shall be Fdi.<version>.nupkg. The FDI® Type Library shall be versioned as per IEC 62769-1:2021, 8.1. The FDI® Type Library is part of the FDI® Core Technology as per IEC 62769-1:2021, 8.3.2.1. Therefore, it directly influences the FDI® Technology Version. All compatible changes of the FDI® Type Library lead to an increase of the minor portion of the FDI® Technology Version. Incompatible changes lead to an increase of the major portion of the FDI® Technology Version (see IEC 62769-1:2021, 8.3.2.2). The version information of the FDI® Type Library can be found in FCG TS10099.

The FDI® Type Library is signed with a single unique key by the issuer of the file. The FDI® Type Library shall be installed separately as part of every FDI® Client installation. User Interface Plug-Ins (UIP) and the FDI® Client Application shall use this instance of the FDI® Type Library. UIPs shall not carry or deploy the FDI® Type Library. The FDI® Client is responsible to provide means to allow updates of this type library over time.

Figure 1 shows the FDI® Type Library structure.

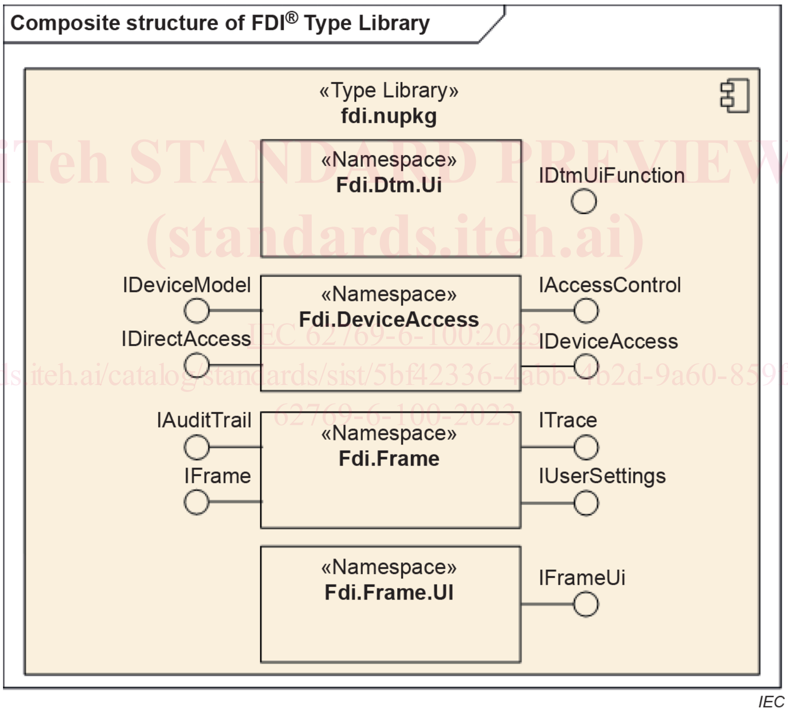

Figure 1 – FDI® Type Library structure

NOTE The composite structure diagram shows only the core interfaces that implement the interfaces defined in IEC 62769-2.

4.2 UIP representation

The UIP Variant can contain either a single or multiple runtime modules (.NET Assembly) and their related supplementary files (for example: resource files). The runtime module of the UIP Variant is called UIP executable. The supplementary file(s) of the UIP Variant is/are called UIP supplement(s).

UIP supplement(s) is/are stored under (a) subfolder(s) of the UIP executable installation directory.

EXAMPLE Examples of UIP supplementary data files include resource files and application configuration data.

The supported RuntimeIds and .NET Framework versions for a specific FDI® Technology Version are specified in FCG TS10099 FDI® Technology Management.

The UIP Variant shall be self-contained. All UIP required libraries (.NET Assemblies) required by a UIP Variant are stored within the same Folder.

4.3 UIP executable representation

The implementation of the UIP depends on the type of user interface elements that can be embedded into the user interface hosting environment of the FDI® Client. UIP shall be implemented as a .NET `System.Windows.Forms` class `UserControl` or a WPF `System.Windows.Controls` class `UserControl`.

UIP executables and their required libraries shall have strong names. The signing of a strong named Assembly can be done using a self-generated key.

NOTE The identity of strong named Assemblies consists of a name, version, culture, public key token and digital signature.

UIP executables and their required libraries shall be shipped with file containing the public key in order to enable Assembly verification.

4.4 UIP executable compatibility rules

The compatibility rules for different versions of the UIP component are specified in IEC 62769-4.

The compilation target platform for the UIP shall be "anyCPU". If this is not feasible the UIP shall be shipped in two variants. One UIP variant shall be compiled for target platform "x86". The second UIP variant shall be compiled for target platform "x64". The compilation platform target shall be described in the catalog.xml file which is defined in IEC 62769-4. This catalog.xml file contains an xml element "CpuInformation" that describes the User Interface Plug-in variant. The allowed values that shall be used in the xml element "CpuInformation" are "anyCPU", "x86" or "x64".

4.5 Allowed .NET CLR versions

4.5.1 General

Specific CLR versions are released for the execution of software components built with specific .NET Framework versions. The .NET CLR version 4.0 is used to execute software components built with .NET Framework 4.0. .NET Components are built for one CLR version only but can be able to run also under a newer CLR version.

FDI® Clients can be built based on CLR version 4.0 or future versions. An FDI® Client has to realize the following situations when starting a UIP.

- When the UIP to be started was built for the same run-time, the UIP can be started in the FDI® Client as usual.
- When the UIP to be started was built with another CLR version and is not compiled for the current running CLR version, the FDI® Client shall start the UIP in a surrogate process with the adequate CLR version. (More details are described in 4.5.2.)

Taking this behavior in account, a UIP shall be developed for CLR version 4.0 or any future version. In case the CLR versions do not match, the UIP shall be started in a separate process. The UIP will then not be displayed as an integrated module within the FDI® Client. It is up to the FDI® Client to realize the surrogate process.

4.5.2 CLR compatibility strategy

In the future, FDI® Clients and UIPs will be permitted to be built on different incompatible versions of the CLR.

If an FDI® Client detects that a UIP requires a CLR that is not compatible with the FDI® Client, the FDI® Client can use a proxy class that enables interaction with the UIP built using a different version of the CLR.

The FDI® Client loads a proxy UIP executable, creates an instance of the proxy class, and delegates the execution of the UIP to this proxy. The proxy starts a process with the required CLR and executes the UIP in this surrogate process. The proxy classes provide the standard FDI® interfaces. The FDI® Client can use these interfaces to interact with the UIP executed in the surrogate process.

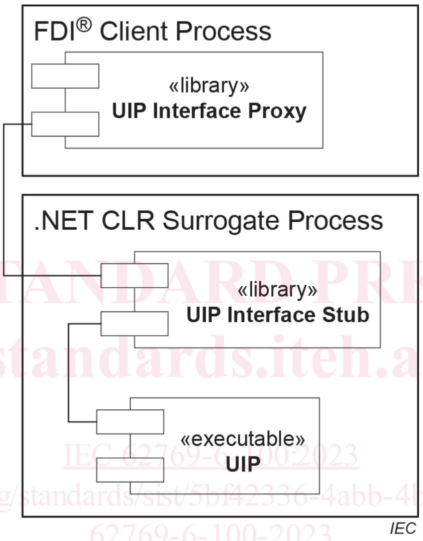

Figure 2 – .NET surrogate process

4.5.3 How to identify the .NET target platform of a UIP

The .NET target platform CLR version information for which a certain Assembly is compiled can be extracted by means of .NET Framework library functions (see Figure 3).

```
clrVersion = Assembly.LoadFrom(<Assembly Path>).ImageRuntimeVersion;
```

IEC

Figure 3 – Identification of Run-time Version

NOTE The Visual Studio®[2] 2008 and 2010 IDE allow developers to select the .NET Framework target. The selection of a .NET Framework target older than the base for the current Visual Studio® IDE automatically creates a configuration file listed as "app.config" within the solution explorer. This file only reflects the current complier setting. The compiler does not read that file.

[2] Visual Studio is the trademark of Microsoft Corporation. This information is given for the convenience of users of this document and does not constitute an endorsement by IEC of the trademark holder or any of its products. Compliance does not require use of the trademark. Use of the trademark requires permission of the trademark holder.

4.6 UIP Deployment

The general UIP installation rules are outlined in IEC 62769-2. The UIP executable shall not be registered within the Global Assembly Cache.

The "strong-name" rule ensures that related Assemblies of different versions of the UIP can coexist during runtime.

The FDI® Client implementation ensures that UIP deployment works independently from current user credentials. (See the NOTE below.)

NOTE Certain operating system managed folders require specific access rights, for example, modifications in folder "Program Files" require "Administrator" rights. The Windows operating system provides several means to allow an application running with restricted user rights, to execute actions with administrator privileges transparent to the user, for example, special restriction handling for identified directories, services with administration rights, executables that are configured to automatically run with administration rights. The alternative is to copy UIP executables into folders writeable for "normal" users.

4.7 UIP Life-cycle

4.7.1 General

The UIP state machine, outlined in IEC 62769-2, is composed of the Loaded, Created, Operational, Deactivated and Disposed states. The mechanisms affecting state changes are described in 4.7.

After the FDI® Client has stored the UIP executable on the FDI® Client the FDI® Client loads the UIP Assemblies dynamically into the memory and executes the related logic by calling the corresponding FDI® specified interface functions.

Subclause 4.7 describes rules about how the FDI® Client shall activate and deactivate the UIP.

4.7.2 UIP Assembly activation steps

4.7.2.1 Load

The FDI® Client shall load the UIP executables by using the LoadFrom mechanism. The .NET framework provides System.Reflection.Assembly.LoadFrom for this purpose:

The LoadFrom mechanism behaves as follows.

- LoadFrom loads the Assembly addressed with the file path and also the referenced Assemblies located within same directory. The argument string assemblyFile shall contain the file name of the UIP executable. The file name of the UIP executable represents the StartElementName described in IEC 62769-4.
- If an Assembly is loaded with LoadFrom, and later an Assembly in the "load context" attempts to load the same Assembly by display name, then this load attempt fails.
- If an Assembly with the same identity is already loaded (for example, by another UIP), then LoadFrom returns the Assembly that has been loaded before, even if a different file path was specified. Even a different file name does not matter. Only the identity of the Assembly is relevant.
- If an Assembly is loaded with LoadFrom, and the probing path includes an Assembly with the same identity (for example, in the Global Assembly Cache or an application directory), then this Assembly is loaded, even if a different file path was specified.
- `LoadFrom` requires the permissions `FileIOPermissionAccess.Read` and `FileIOPermissionAccess.PathDiscovery`, or `WebPermission`, on the specified path.
- `LoadFrom` loads the assembly into the default Application Domain.

- If a native Assembly image (generated by ngen.exe) exists for the specified file path, then it is not used. The Assembly cannot be loaded as domain neutral, i.e., the Assembly cannot be shared between Application Domains.

This behavior enforces deployment rules as follows.

- Rules regarding Assembly dependencies (see 4.7.2.4.2).

The FDI® Client shall only use `LoadFrom`. The use of other .NET Assembly loading/object creation means is not allowed.

- Rules regarding shared Assemblies (see 4.7.2.4.3).
- A pre-compiled processor-specific machine code cannot be used.
- The security aspects regarding loading and execution of Assemblies are described in 4.9.

4.7.2.2 Create

Creating an instance of the UIP Assembly works using the .net library functions `System.Reflection.Assembly.GetTypes` and `System.Activator.CreateInstance`. The FDI® type library declares a "custom attribute" named `UIPActivationClass`. This attribute shall only be added to the object implementing the interface `IDtmUiFunction` that actually implements the UIP start-up function. The attribute `UIPActivationClass` shall be used once only.

The FDI® Client can now use `System.Reflection` services to clearly determine the UIP implemented activation procedure.

NOTE 1 Function System.Reflection.Assembly.GetTypes can be used to query the interface IDtmUiFunction.

NOTE 2 Function System.Attribute.GetCustomAttributes can be used for reading the additional custom attributes.

NOTE 3 The result of function invocation System.Activator.CreateInstance is an object of type IDtmUiFunction.

A data type cast is needed.

4.7.2.3 Activate

Invocation of function `IDtmUiFunction.Init` finally activates the UIP for the user.

4.7.2.4 External libraries

4.7.2.4.1 General

UIP Assemblies can depend on external libraries (3rd party libraries) and other Assemblies, for example, specific user control libraries. FDI® Clients do not perform installation of UIPs, rather they dynamically load and execute the UIP. To support this usage, as well as the requirement to prevent possible problems of conflicting Assemblies, rules are specified for external libraries.

External libraries shall:

- be contained within the FDI® Package;
- not require Microsoft Installer (MSI) installation;
- not require entries in the Windows Registry or the Global Assembly Cache;
- adhere to the access restrictions described in 4.9.2;
- be compatible with the platforms described in IEC 62769-6.

4.7.2.4.2 Loading of external libraries

The FDI® Client loads the UIP Assembly, containing the UIP main class implementing interface `IDtmUiFunction`, by invocation of the .NET framework function `LoadFrom`. Referenced Assemblies that are stored in the same directory are automatically loaded together with this .NET Assembly. Referenced Assemblies that are stored in other locations (for example, in a sub-directory) have to be loaded explicitly by the UIP itself.

The UIP shall load such Assemblies also by invocation of the .NET framework function `LoadFrom`. Loading Assemblies with other .NET framework methods is not allowed.

Usage of external libraries shall not break the self-containment requirement for FDI® Packages; all external libraries shall be included in the FDI® UIP Package

4.7.2.4.3 Loading of shared external libraries

An external library is a shared external library if a related .NET Assembly identity can be used from different UIP executables. The identity of a .NET Assembly matters. Installation path and Assembly filename are not relevant.

Usage of shared libraries shall not break the self-containment requirement for FDI® Packages. Each of the delivered FDI® Packages shall be shipped with all required UIP related libraries. The sharing mechanism comes from the .NET framework implemented optimization mechanism.

If a shared Assembly is used, then the following rules apply.

- Any incompatible change to the shared Assembly shall lead to a new identity, for example, different version number.
- Shared Assemblies shall not presume to be loaded from a specific installation path, for example, rely on the fact that some files are stored in the same directory or in a sub-directory.
- Static variables in shared Assemblies are also shared if the Assembly is loaded into the same Application Domain. Thus static variables shall not have side effects in such scenarios. External shared libraries shall not declare static variables.
- Because of the self-containment rule defined for the FDI® Package, shared Assemblies shall be deployed with all FDI® Packages using a shared Assembly.

4.7.2.5 UIP Constructor invocation

Constructor and destructor implementation shall not throw exceptions. The constructor logic shall be limited to instantiate the object in terms of the internal data structure. The destructor logic shall be limited to destroy the object in terms of releasing memory resources. The constructor and the destructor shall not:

- Invoke any call-back to the FDI® Client.
- Invoke any user interaction.

4.7.3 UIP Assembly deactivation steps

4.7.3.1 Deactivate

For UIP deactivation the FDI® Client shall call the interface `IDtmUiFunction.BeginClose` and `IDtmUiFunction.EndClose`. On successful execution the UIP shall release all resources and the FDI® Client shall delete all references to the UIP instance. The .NET garbage collector finally disposes the UIP runtime object.