## **Introduction**

## This article provides a brief overview of technology in MicroStation CONNECT …

## [Item Types](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#ItemTypes)

## [EC Schemas](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#EcSchemas)

## [Class Editor](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#ClassEditor)

## The information presented here is an informal guide to *Item Types*. Its intended audience includes MicroStation administrators and MicroStation developers.

## **Programmer's Notes**

## *Item Types* is the term used in user documentation. The *MicroStationAPI* and *DgnPlatformNet* APIs uses terms such as CustomPropertyContainer.

## VBA support for *Item Types* arrived with MicroStation CONNECT Update 10. See the following class descriptions in VBA help …

## ItemType

## ItemTypeProperty

## ItemTypePropertyHandler

## ItemTypeLibrary

## ItemTypeLibrary

## AreaAnnotator™

## This article about [AreaAnnotator CONNECT Edition](http://www.la-solutions.co.uk/content/connect/Products/AreaAnnotator/AreaAnnotator.htm)™ shows how *Item Types* help obtain reports about area features in a DGN model.

## [AreaAnnotator CONNECT Edition](http://www.la-solutions.co.uk/content/connect/Products/AreaAnnotator/AreaAnnotator.htm)

## **Feature Table**

## There's a useful table in MicroStation CONNECT help that compares various annotation technologies, including *Item Types* and *Text Fields*. To find this table in MicroStation help, search for *Comparing Text Field, Label, Note, Table, Report, Item, Item Type, and Item Sets*.

## MicroStation CONNECT help is available on line. Here's that [comparison table in MicroStation Help](https://docs.bentley.com/LiveContent/web/pub.xql?c=t&action=home&pub=MicroStation%20Help-v9&lang=en#docid=GUID-0A5C6CFC-E9F6-4F0D-BE00-6F24BE05839F&addHistory=true&query=&scope=&tid=&filename=GUID-0A5C6CFC-E9F6-4F0D-BE00-6F24BE05839F.xml&resource=&inner_id=&toc=false&eventType=lcContent.loadDocGUID-0A5C6CFC-E9F6-4F0D-BE00-6F24BE05839F&url=/LiveContent/web/search.xql%3Fc%3Dt%26pub%3DMicroStation+Help-v9%26lang%3Den%26action%3Dsearch%26query%3DComparing%2520Text%2520Field&sid=lcSearch.runSearch1510684824152&currentQuery=Comparing%2520Text%2520Field&currentScope=).

## Item Sets

## To dispel confusion, [Item Sets](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#ItemSets) are an earlier and different technology. They are unrelated to *Item Types* other than having a similar name.

## **Item Types**

## *Item Types*, new to MicroStation CONNECT, are meta data that define data types. In other words, a schema. An *Item Type* has a name and a number of properties. Each property has a name and a data type (e.g. *Text*, *Integer*, *Number*). *Item Types* are not graphic elements; rather, they represent a MicroStation capability for storing structured alpha-numeric data on an object. *Item* data can be attached to an element, DGN model or DGN file. Bentley Systems terminology for data stored in *Item Types* includes 'business data' and 'EC Schema data'.

## *Item Types* define data, not graphics

## A graphic element can be associated with *Item Type* data

## MicroStation CONNECT provides tools to enable a user to make that association

## A text field can link to *Item Type* data

## Hence text elements, labels and notes can display *Item Type* data

## *Item Types* are the data source for *Reports*, new to MicroStation CONNECT

## *Reports* are the precursor for *Tables*, also new to MicroStation CONNECT

## *Item Types* provide a simplified way for users and applications to create, manage, and use EC data within DgnFiles.

## An *Item Type* is an ECClass which supports a subset of the features available to ECClasses. In comparison to full-fledged ECClasses, *Item Types*:

## Do not support inheritance

## Provide a default value for every primitive property

## Do not support most custom attributes

## Support only a portion of available property types CustomProperty::Type

## Can be used by *Item Type*-based tools, such as MicroStation's *Attach Item* tool

## *Item Types* are stored an *Item Type Library*, which is a similarly restricted ECSchema. In comparison to full-fledged ECSchemas, *Item Type Libraries*:

## Can reference only standard schemas (chiefly for custom attributes)

## Do not support most custom attributes

## Cannot define custom attribute classes

## Can be freely modified, deleted, updated, and renamed by the user

## An *Item Type Library* is a resource, stored in a DGN file. Typically they are managed by defining the *Item Type* and its properties in a DGNLib, which is made available to all users who need to use that *Item Type*. A management dialog is provided by MicroStation for the end user …

## An *Item Type Library* is identified by its name (usually provided by the user)

## A DgnFile which uses an *Item Type Library* stores a copy of that library locally

## A local *Item Type Library* can be replaced with an updated version from a dgnlib

## When copying elements between files, if the element references an *Item Type Library*, it will be remapped to use an existing *Item Type Library* by the same name in the destination file; or else the *Item Type Library* will be copied into the destination file.

## *Item Types* are metadata. They become useful when they are "attached" to a host to produce an *Item*. An *Item* is an ECInstance of an *Item Type* ECClass.

## **New to MicroStation CONNECT Update 10**

## From MicroStation CONNECT Update 10 *Item Types* are supported by MicroStation VBA.

## MicroStation CONNECT Update 10 introduces the *Picklist*. A *Picklist* is a user-defined list of possible values for an *Item* property.

## For example, suppose you want to annotate door cells with your *Item Type*. One property of your *Item Type* is *door-type*. You would like to restrict the value of the *door-type* property to *internal door*, *internal fire door*, *external door* and *fire escape door*. A *Picklist* enables you to constrain what the user can assign to your *Item*'s *door-type* property.

## **MicroStation CONNECT Update 7**

## MicroStation CONNECT Update 7 introduced the [Edit Item Dialog](https://docs.bentley.com/LiveContent/web/pub.xql?c=t&action=home&pub=MicroStation%20Help-v9&lang=en#docid=EditItem&addHistory=true&query=&scope=&tid=&filename=EditItem.xml&resource=&inner_id=&toc=false&eventType=lcContent.loadDocEditItem).

## 

## **Reports**

## Reports Icon

## MicroStation CONNECT introduced *Reports*. *Reports* is an analytic tool. It extracts alphanumeric data (business data) from your DGN file, model, elements or elsewhere. When *Item Types* are in use, *Reports* can extract *Item* instance data attached to your DGN model or elements.

## Visit the [reports page](http://www.la-solutions.co.uk/content/connect/Reports/Reports.htm) for more information, and to see an example of *Reports* in action.

## **Contributors**

## I am indebted to Bentley Systems staff member Paul Connelly and to [Be Communities MVP](http://communities.bentley.com/p/bentleyempoweredmostvaluableprofessionals) Jan Slegr. Both have contributed both factual information about *Item Types* and *EC Schemas* as well as their interpretation of how they can be used. They contribute enthusiastically to the [MicroStation Programming Forum](http://communities.bentley.com/products/programming/microstation_programming/f/343173).

## **EC Schemas**

## *EC Schemas* were introduced with MicroStation V8*i* and are continued in MicroStation CONNECT. *EC Schemas* are meta data that define data types, analogous to a relational database schema. *EC Schemas* are not graphic elements; rather, they represent a MicroStation capability for storing structured alpha-numeric data. Bentley Systems terminology for data stored in *Item Types* include 'business data' and 'EC Schema data'.

## *EC Schemas* define data, not graphics

## An *EC Schema* defines application-specific business data

## A graphic element is often associated with *EC Schema* data

## MicroStation does not provide tools to enable a user to make that association

## The association is created by an application based on MicroStation (e.g. AECOSim)

## *Reports*, as provided by MicroStation CONNECT, do not work with *EC Schema* data

## The MicroStation ECX command lets you [export a schema](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#ExportSchema), or other information.

## Q To define an Item Type Library implicitly defines an EC Schema?

## Yes.

## Q Is the schema that it defines a first-class schema, equivalent to a schema defined any other way?

## Yes, to workflows which operate on *EC Schema*s, Item Type Libraries look like any other *EC Schema*. e.g., Reporting, Explorer, etc.

## Q That is, do we lose any capability by creating an Item Type Library?

## You lose some capabilities and gain others.

## An *Item Type Library* supports a subset of the features supported by hand-rolled *EC Schema*s, for two reasons:

## We wanted to simplify some EC concepts so the average user can comprehend and use them (if you have used Class Editor, you have some inkling of how complicated *EC Schema*s can become).

## We needed to simplify things in order to design a usable user interface around editing schemas (I do not consider Class Editor a user-friendly interface).

## At the time of writing with respect to MicroStation CONNECT Update 6, some EC features are not supported by *Item Types* …

## Class inheritance

## Nested structs

## ECRelationship classes

## Calculated properties (very much desired though)

## Read-only properties

## Non-standard schema references (i.e. each Item Type Library is self-contained)

## Some capabilities are only available for *Item Types* …

## Attach/Detach Item tool

## Ability to create/edit within MicroStation

## Ability to import from dgnlibs

## Integration with parametric modeling via Variables dialog

## Neither of the above is a comprehensive list.

## Q To define an Item Type Library implicitly defines an *EC Schema*?

## It can be discovered easily:

## Create a new empty file.

## Use ecx schema list key-in to list all schemas stored in the file.

## Create your *Item Type* definition and save it to the file.

## Use the key-in again to check how the number of stored schemas changed.

## You will see some standard (supplement) schemas area added and also DgnCustomItemTypes\_<name>.01.00 schema is created.

## For further research you can [export the schema](http://www.la-solutions.co.uk/content/connect/ItemTypes/ItemSets-ItemTypes-ECSchemas.htm#ExportSchema) and open it in Bentley Class Editor.

## **Export ECSchema**

## There a number of MicroStation key-in commands that enable export of an ECSchema or a sub-set of a schema …

## ECX SCHEMA EXPORT *schema-name*

## ECX SCHEMA IMPORT *schema-name* ! Use with caution !

## ECX SCHEMA LIST

## ECX SCHEMA UPDATE

## ECX ITEMS COUNT

## ECX ITEMS DUMP

## Export Schemas Dialog

## Those ECX commands have some options …

## [help|?]

## Display help message

## [output\_filepath\_with\_no\_spaces]

## The full path and name of the file to which the XML dump is written. There can be no spaces in the path or file name. If this is not supplied, the dump is generated to the user's TEMP folder with a unique name derived from the Active Design File Name.

## [Verbosity=Full|Minimal]

## Optional parameter that controls the level of information. The default is *Full*, which will dump fully serialized instance data. *Minimal* will only output IDs and class names.

## [InstanceType=All|Intrinsic|Extrinsic]

## Optional parameter that controls what types of instances are dumped. The default is *All*. *Extrinsic* refers to business data stored on elements/models, e.g. ECXA instances. *Intrinsic* refers to instances representing intrinsic properties of elements/models/files, e.g. the geometry of a line element.

## **Class Editor**

## About the time that MicroStation CONNECT Update 5 was released, Bentley Systems announced the first CONNECT *ECSchema Editor 1.0*. The title *editor* belies the fact that is a read-only viewing tool. Perhaps later versions will enable editing.

## None the less, it provides a useful way to view the structure of a schema. Moreover, its help documentation provides a valuable insight into the *raison d'être* of ECSchemas and their use in products from Bentley Systems.

## Bentley Systems developer Paul Connelly commented about *Class Editor V8* …

## Don't edit an *Item Type Library* schema in Class Editor V8!

## Veteran independent MicroStation developer Jan Slegr commented …

## I am not sure if there is "Bentley Class Editor CONNECT Edition" available. Checking my computer with CONNECT Edition SDK installed I can found Class Editor 8.1, which I guess was installed together with Navigator or V8i SDK (not sure ;-)

## It would be nice if Class Editor will be enhanced. If I need to create some schema (which is not very often), I am quite happy with the current version, but:

## From time to time it crashes or starts work in a strange way. Restart the app is quick but annoying.

## Very limited documentation about e.g. available Property Attributes ro more advanced concepts like relationship classes etc. This is not primarily Editor issue, but it limits its usage.

## Jan further added *My personal list* …

## Based on stable powerfull EC Framework technology. Maybe it's an "internal thing", but EC data is common and used by many Bentley products and it's stored as XAttributes, so it's fully supported and compatible with DGN V8, i-model and Bentley world in general. Not as tags, that were implemented "on top of DGN V7" with many limitations

## Fully compatible with i-Model, so nothing is lost when exported e.g. to Navigator and used on Android or iPad (including ability to search

## Can be attached not to elements only, but also to design while, to particular model and also to references and attached rasters

## Easy to search in Explorer for Item Types including standard table display format (including ability to export the results to CSV or Excel)

## Possibility to create (and store) complex queries

## *Item Types* can be used as an input to other tools (reports, charts etc)

## Standard storage format allows to access data easily by other applications

## **Item Sets**

## We don't discuss *Item Sets* here in any detail. We mention them because of the similarity in their name to *Item Types*. *Item Sets* were introduced with MicroStation V8*i*. *Item Sets* are unrelated to *Item Types*.

## *Item Sets* are a superset of MicroStation *Named Groups*

## They have searching and filtering tools

## They can be associated with *EC Schema* Data

## *Item Sets* are not a data source that can be used by labels or notes via *text fields*