Introduction to Plex Meta Coding

Morten Knudsen, Soft Design

2E/PLEX plex2e.com

2013 WORLDWIDE DEVELOPER CONFERENCE

Morten Knudsen

Soft Design, Consultants

Do you know about Plex specification and development, and have you heard about *Meta coding*? This presentation gives a short introduction to Plex *Meta code* facilities making Plex a unique and strong model-based development tool. In addition to a *Meta code* overview and basic concepts, this presentation will cover practical examples as well as a discussion on what kind of challenges that can be solved with *Meta coding*.

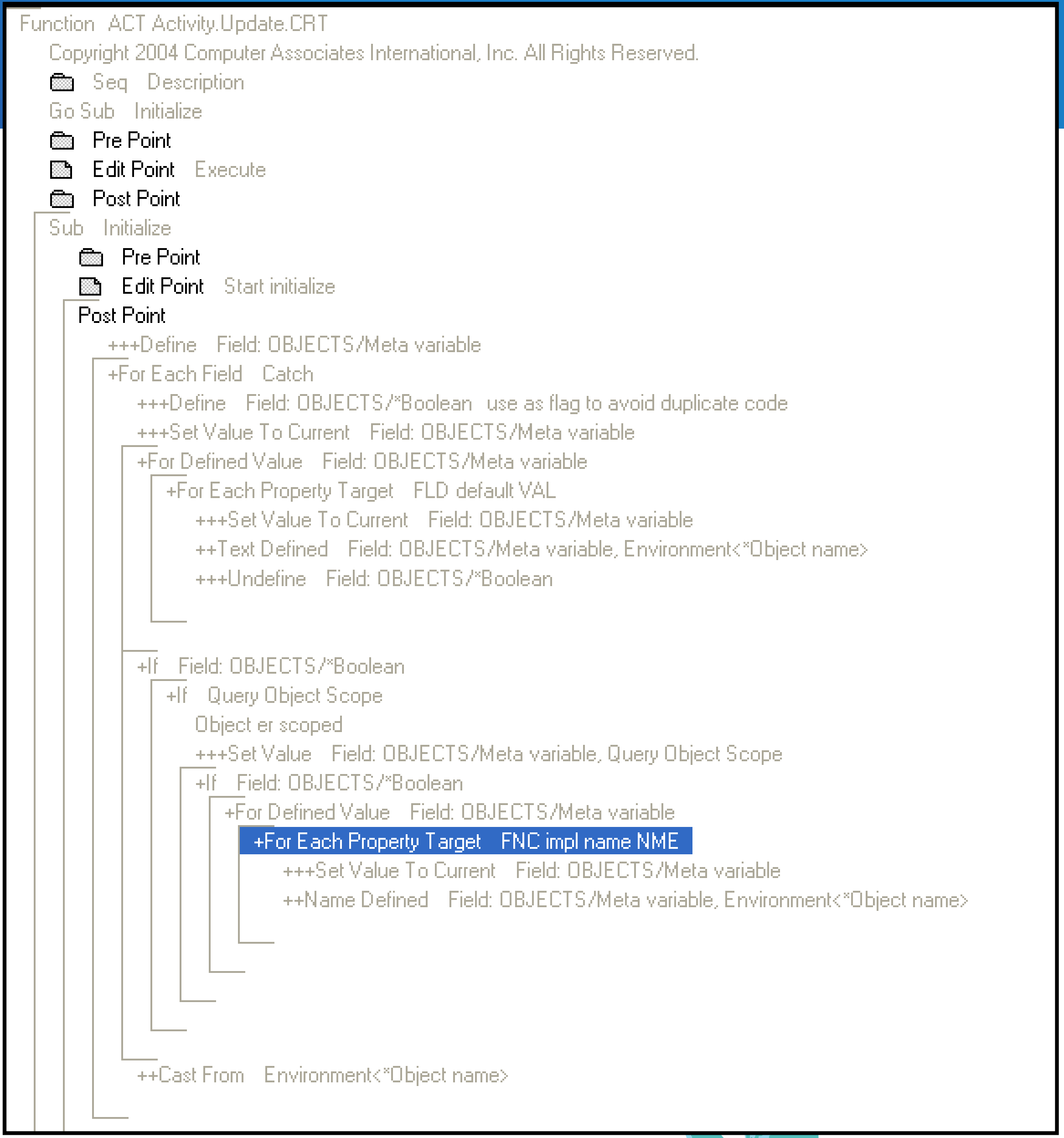
Agenda

- Introduction to Plex Meta Coding
- Meta Teasers – Small Examples
- Specific Problems Solved by Meta
- What Can Be Done with Meta?
- Working with Meta Coding
- Questions and Answers

Introduction to Meta Coding

What Is Plex Meta Code (or just Meta)?

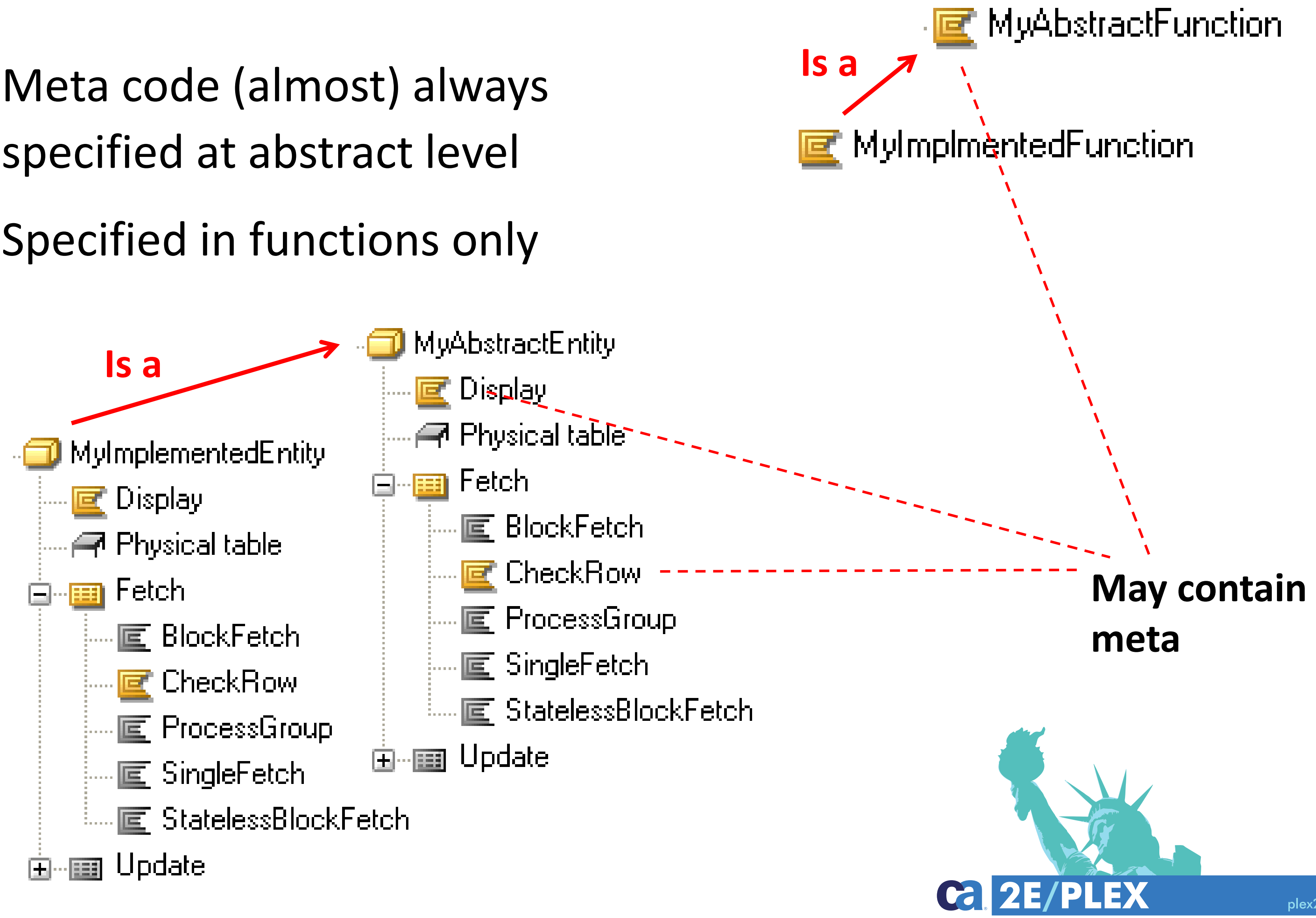
- Ever wondered about the “+” statements in inherited code?
- Meta provides a way to control generation of code!



```
Function  ACT Activity.Update.CRT
  Copyright 2004 Computer Associates International, Inc. All Rights Reserved.
     Seq   Description
  Go Sub   Initialize
     Pre Point
     Edit Point   Execute
     Post Point
  Sub   Initialize
       Pre Point
       Edit Point   Start initialize
    Post Point
      +++Define   Field: OBJECTS/Meta variable
      +For Each Field   Catch
        +++Define   Field: OBJECTS/*Boolean   use as flag to avoid duplicate code
        +++Set Value To Current   Field: OBJECTS/Meta variable
        +For Defined Value   Field: OBJECTS/Meta variable
          +For Each Property Target   FLD default VAL
            +++Set Value To Current   Field: OBJECTS/Meta variable
            ++Text Defined   Field: OBJECTS/Meta variable, Environment<*Object name>
            +++Undefine   Field: OBJECTS/*Boolean

        +If   Field: OBJECTS/*Boolean
          +If   Query Object Scope
            Object er scoped
            +++Set Value   Field: OBJECTS/Meta variable, Query Object Scope
            +If   Field: OBJECTS/*Boolean
              +For Defined Value   Field: OBJECTS/Meta variable
                +For Each Property Target   FNC impl name NME
                  +++Set Value To Current   Field: OBJECTS/Meta variable
                  ++Name Defined   Field: OBJECTS/Meta variable, Environment<*Object name>

        ++Cast From   Environment<*Object name>
```

Meta Code Specified in Abstract Functions

- Meta code (almost) always specified at abstract level
- Specified in functions only



MyAbstractFunction

Is a

MyImplmentedFunction

MyAbstractEntity
- Display
- Physical table
- Fetch
  - BlockFetch
  - CheckRow
  - ProcessGroup
  - SingleFetch
  - StatelessBlockFetch
- Update

Is a

MyImplementedEntity
- Display
- Physical table
- Fetch
  - BlockFetch
  - CheckRow
  - ProcessGroup
  - SingleFetch
  - StatelessBlockFetch
- Update

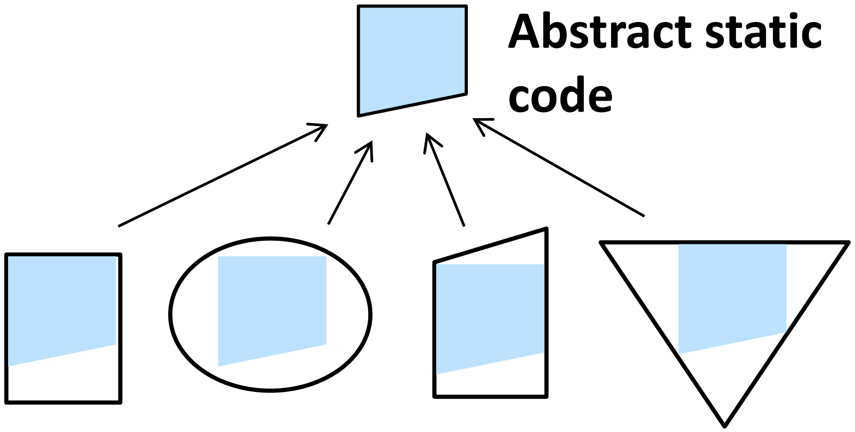
May contain meta

Use Meta to Access Model Triples and Definitions

- Use Meta code to query triples
  - All model triples can be queried by meta code
- Resolve inheritance
  - Inherited triples also queried
- Example of pseudo meta code

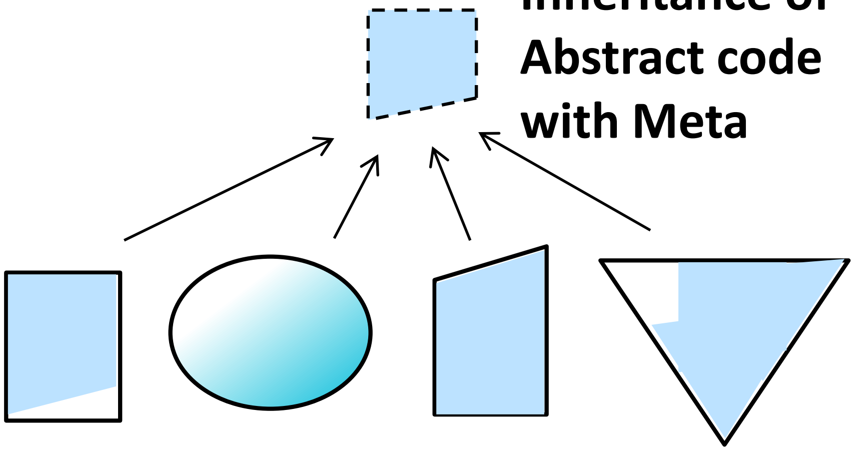
```
+FOR EACH FIELD in specific variable
  +IF field has an initialization function
    ++CALL initialization function
  +IF field is of type Date
    ++IF field is empty/blank
      CALL GetCurrentDate to set value of Date field
```

Context-Dependent Inheritance of Code



Inheritance of Abstract static code

Inherited code (white area to be coded)



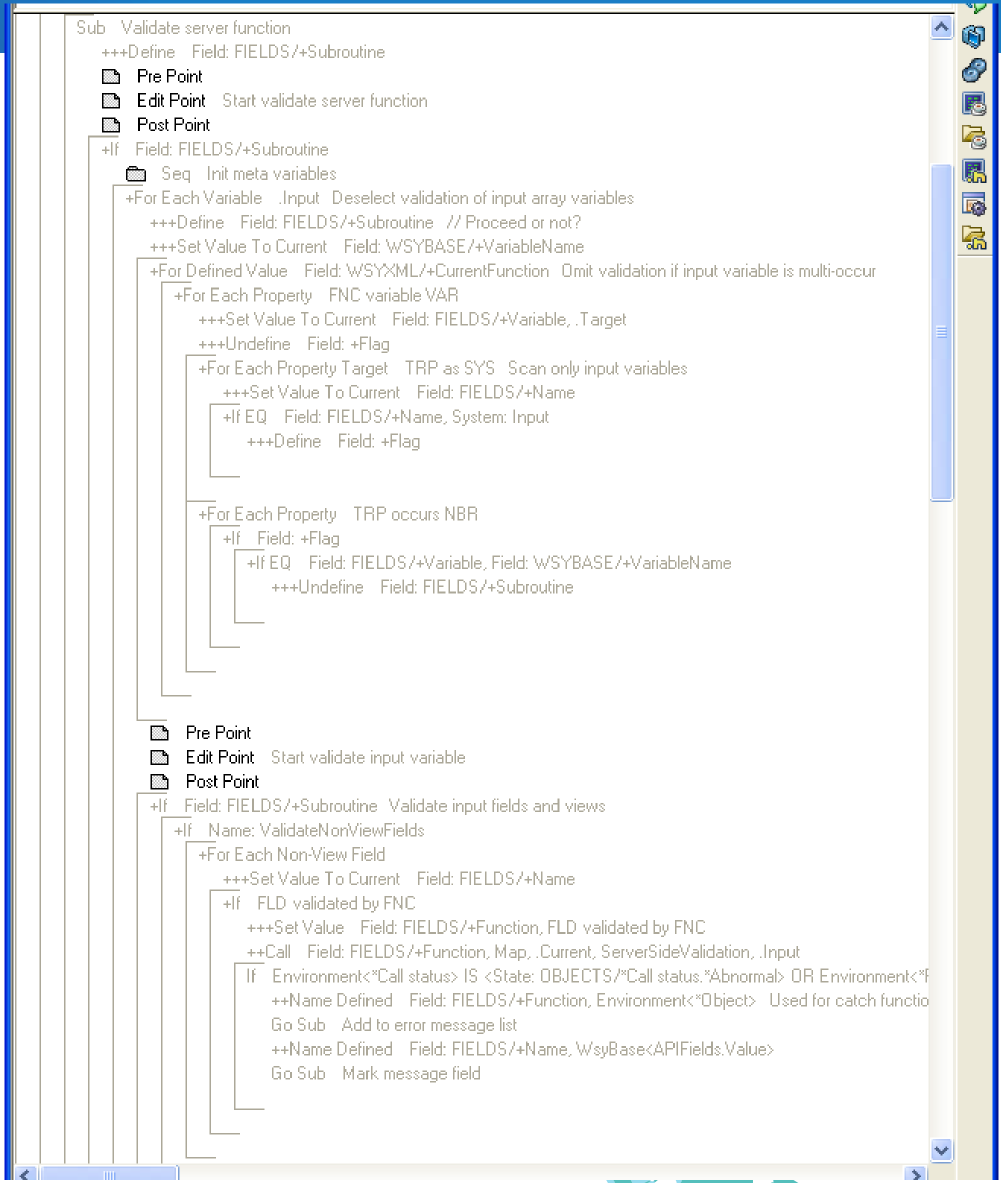
Inheritance of Abstract code with Meta

Inherited code (white area to be coded)

- Inherited code can adjust to context
- Few lines of meta -> Many lines of generated code
- Many lines of meta -> Generate the right line(s) of code

The Unbearable Lightness of Meta Coding

- Works fine, everything is fixed for you
- Draw-back
  - Less transparent for ‘users’ of Meta
  - Harder to debug code in error situations
  - What exactly happened?
  - Where did it go wrong?

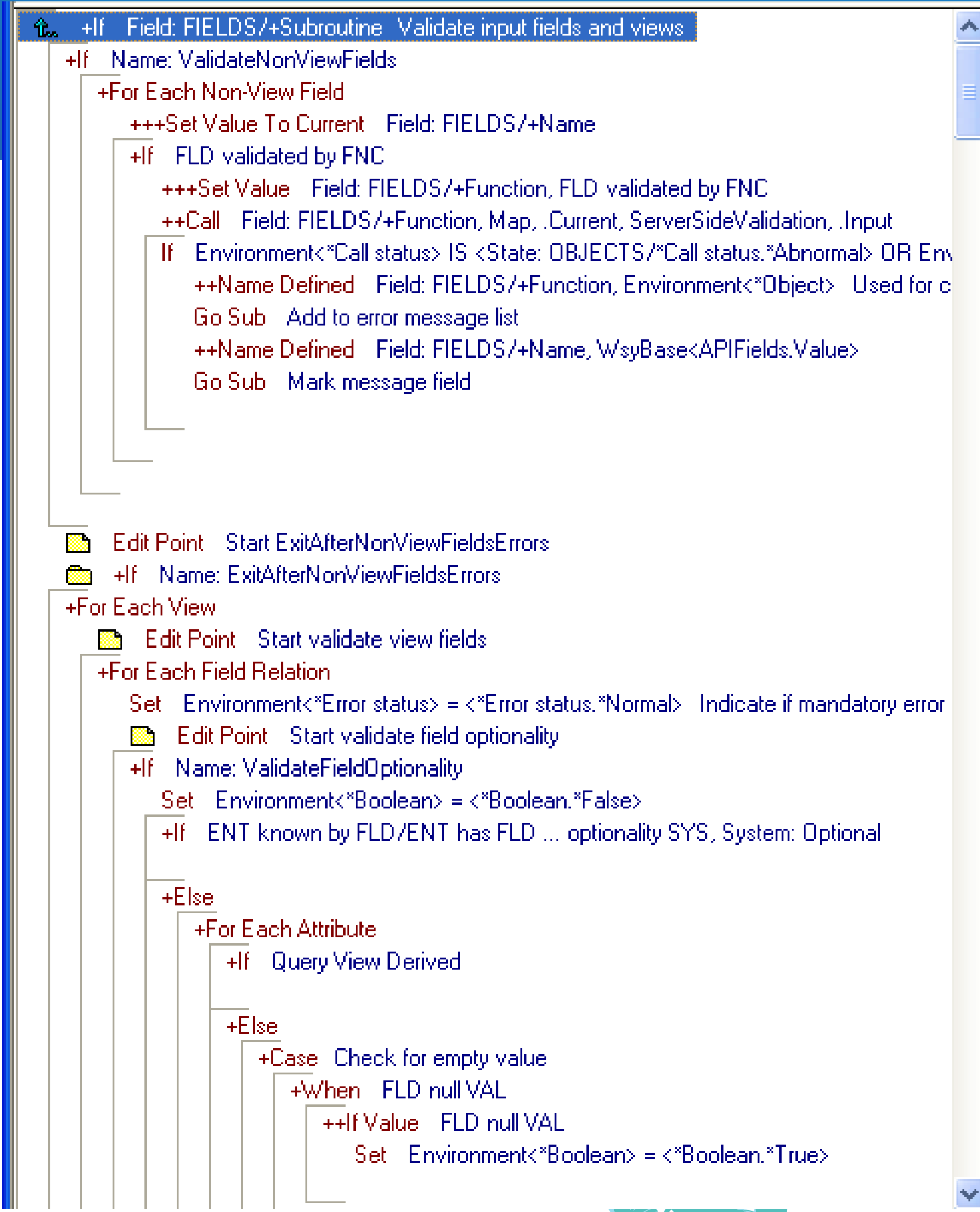


```
Sub   Validate server function
  +++Define   Field: FIELDS/+Subroutine
  Pre Point
  Edit Point   Start validate server function
  Post Point
  +If   Field: FIELDS/+Subroutine
    Seq   Init meta variables
    +For Each Variable   .Input   Deselect validation of input array variables
      +++Define   Field: FIELDS/+Subroutine   // Proceed or not?
      +++Set Value To Current   Field: WSYBASE/+VariableName
      +For Defined Value   Field: WSYXML/+CurrentFunction   Omit validation if input variable is multi-occur
        +For Each Property   FNC variable VAR
          +++Set Value To Current   Field: FIELDS/+Variable, .Target
          +++Undefine   Field: +Flag
          +For Each Property Target   TRP as SYS   Scan only input variables
            +++Set Value To Current   Field: FIELDS/+Name
            +If EQ   Field: FIELDS/+Name, System: Input
              +++Define   Field: +Flag

          +For Each Property   TRP occurs NBR
            +If   Field: +Flag
              +If EQ   Field: FIELDS/+Variable, Field: WSYBASE/+VariableName
                +++Undefine   Field: FIELDS/+Subroutine

      Pre Point
      Edit Point   Start validate input variable
      Post Point
      +If   Field: FIELDS/+Subroutine   Validate input fields and views
        +If   Name: ValidateNonViewFields
          +For Each Non-View Field
            +++Set Value To Current   Field: FIELDS/+Name
            +If   FLD validated by FNC
              +++Set Value   Field: FIELDS/+Function, FLD validated by FNC
              ++Call   Field: FIELDS/+Function, Map, .Current, ServerSideValidation, .Input
              If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnormal> OR Environment<*F
                ++Name Defined   Field: FIELDS/+Function, Environment<*Object>   Used for catch functio
                Go Sub   Add to error message list
                ++Name Defined   Field: FIELDS/+Name, WsyBase<APIFields.Value>
                Go Sub   Mark message field
```

Meta Teasers – Small Examples

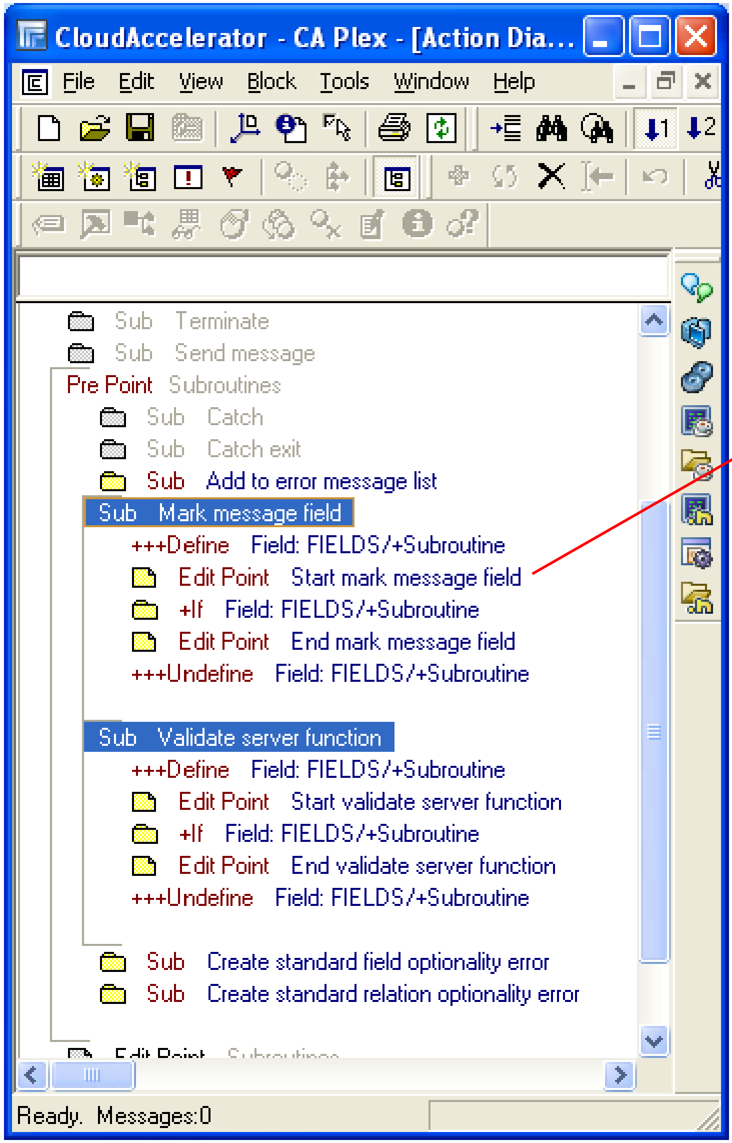
Different Kind of Meta Statements

- "+" Statement
  - Control statements
- "++" Statement
  - Result in generation of code (line)
- "+++" Statement
  - Meta variable definition and assignments
- Meta code often contains 'normal' statements



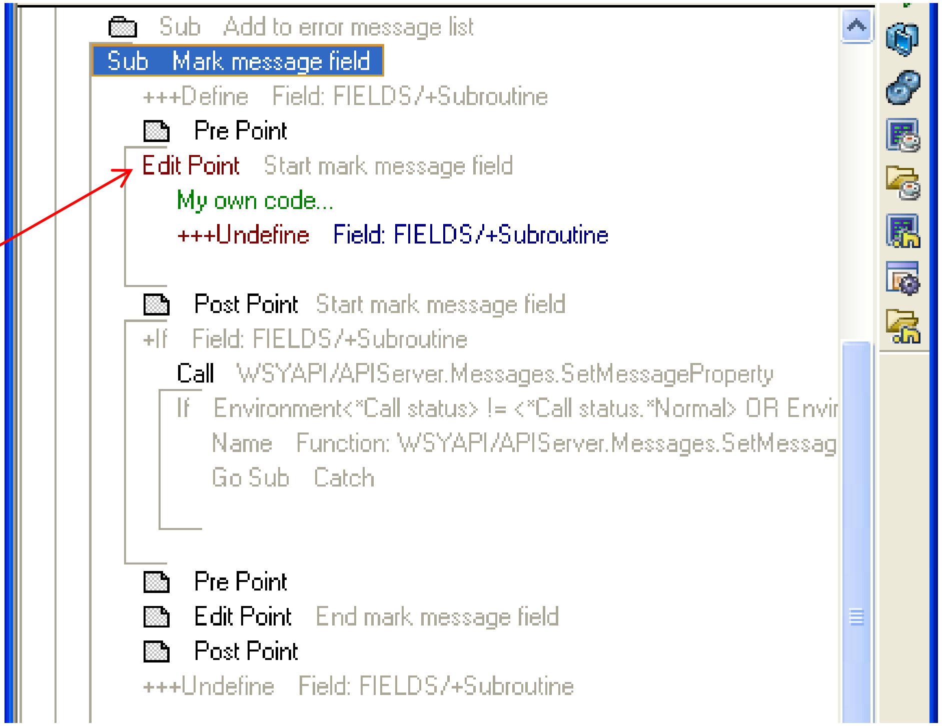
```
+If   Field: FIELDS/+Subroutine  Validate input fields and views
  +If   Name: ValidateNonViewFields
    +For Each Non-View Field
      +++Set Value To Current   Field: FIELDS/+Name
      +If   FLD validated by FNC
        +++Set Value   Field: FIELDS/+Function, FLD validated by FNC
        ++Call   Field: FIELDS/+Function, Map, .Current, ServerSideValidation, .Input
        If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnormal> OR Env
          ++Name Defined   Field: FIELDS/+Function, Environment<*Object>   Used for c
          Go Sub   Add to error message list
          ++Name Defined   Field: FIELDS/+Name, WsyBase<APIFields.Value>
          Go Sub   Mark message field

  Edit Point   Start ExitAfterNonViewFieldsErrors
  +If   Name: ExitAfterNonViewFieldsErrors
  +For Each View
    Edit Point   Start validate view fields
    +For Each Field Relation
      Set   Environment<*Error status> = <*Error status.*Normal>   Indicate if mandatory error
      Edit Point   Start validate field optionality
      +If   Name: ValidateFieldOptionality
        Set   Environment<*Boolean> = <*Boolean.*False>
        +If   ENT known by FLD/ENT has FLD ... optionality SYS, System: Optional

        +Else
          +For Each Attribute
            +If   Query View Derived

            +Else
              +Case   Check for empty value
                +When   FLD null VAL
                  ++If Value   FLD null VAL
                    Set   Environment<*Boolean> = <*Boolean.*True>
```

De-Selectable Subroutines



```
    Sub   Terminate
    Sub   Send message
  Pre Point   Subroutines
      Sub   Catch
      Sub   Catch exit
      Sub   Add to error message list
    Sub   Mark message field
        +++Define   Field: FIELDS/+Subroutine
          Edit Point   Start mark message field
          +If   Field: FIELDS/+Subroutine
          Edit Point   End mark message field
        +++Undefine   Field: FIELDS/+Subroutine

    Sub   Validate server function
        +++Define   Field: FIELDS/+Subroutine
          Edit Point   Start validate server function
          +If   Field: FIELDS/+Subroutine
          Edit Point   End validate server function
        +++Undefine   Field: FIELDS/+Subroutine

      Sub   Create standard field optionality error
      Sub   Create standard relation optionality error

    Edit Point   Subroutines
```

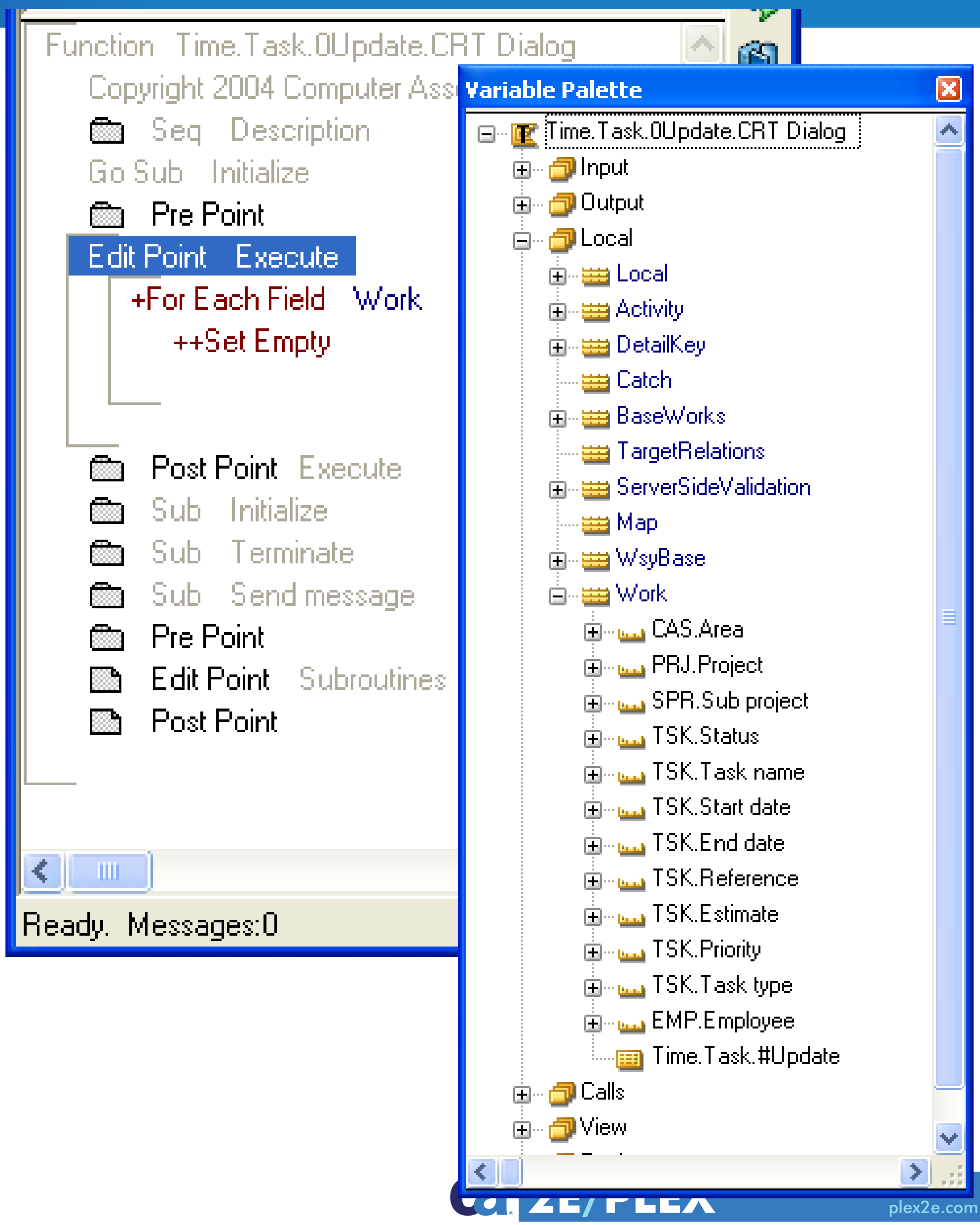
Ready. Messages:0



```
    Sub   Add to error message list
  Sub   Mark message field
      +++Define   Field: FIELDS/+Subroutine
        Pre Point
      Edit Point   Start mark message field
        My own code...
        +++Undefine   Field: FIELDS/+Subroutine

        Post Point   Start mark message field
      +If   Field: FIELDS/+Subroutine
        Call   WSYAPI/APIServer.Messages.SetMessageProperty
        If   Environment<*Call status> != <*Call status.*Normal> OR Envir
            Name   Function: WSYAPI/APIServer.Messages.SetMessag
            Go Sub   Catch

        Pre Point
        Edit Point   End mark message field
        Post Point
      +++Undefine   Field: FIELDS/+Subroutine
```

Clear Value of All Fields in a Variable by ++Set empty

- Fast way to reset all fields in a specified variable
  - Abstract and Implemented
- Alphanumeric fields set to blank
- Numeric fields set to 0
- Generated code depends on context (triples)
  - One line of code generated for each field in local *Work* variable



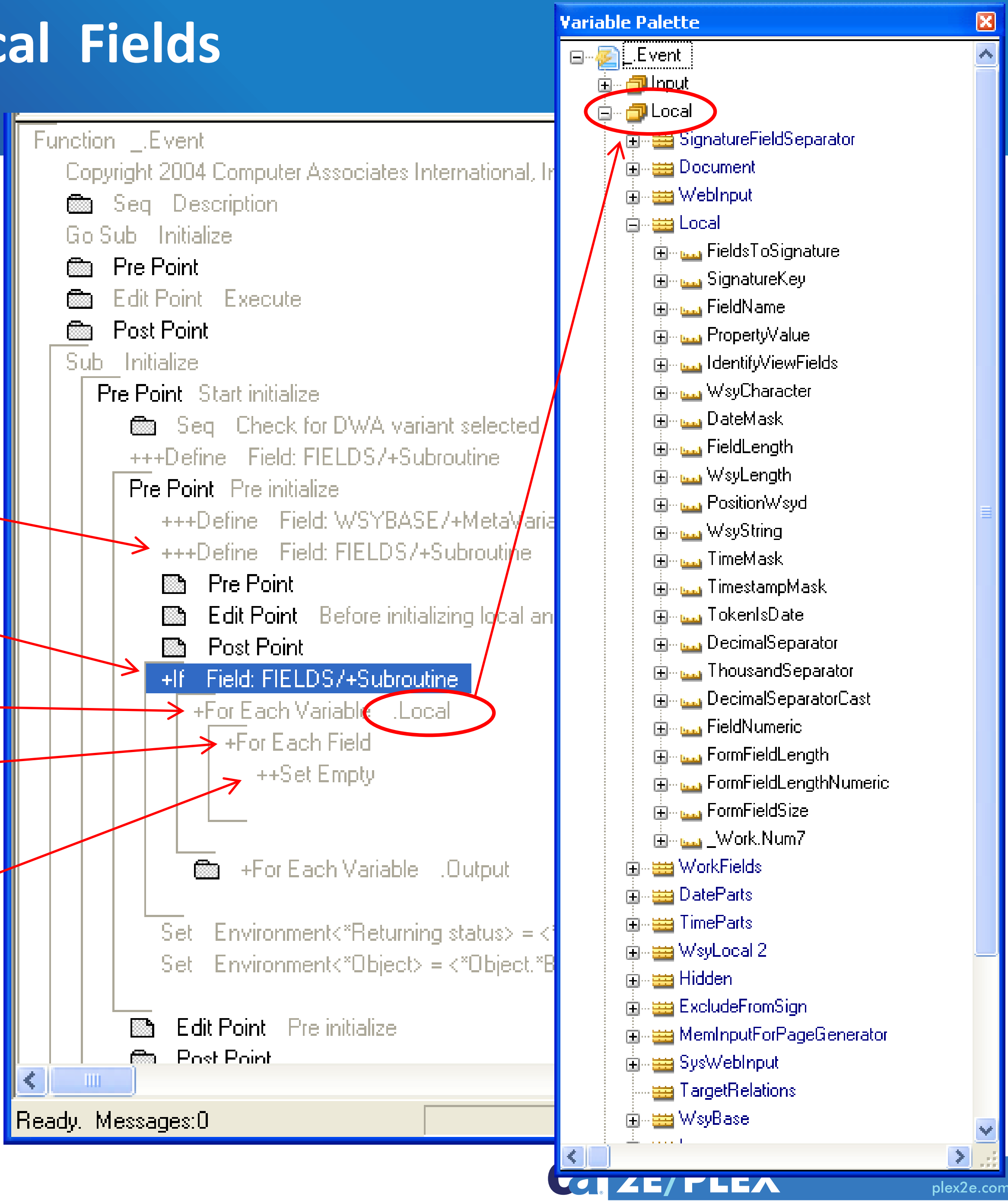
```
Function   Time.Task.0Update.CRT Dialog
    Copyright 2004 Computer Ass
    Seq   Description
    Go Sub   Initialize
    Pre Point
    Edit Point   Execute
       +For Each Field   Work
          ++Set Empty
    Post Point   Execute
    Sub   Initialize
    Sub   Terminate
    Sub   Send message
    Pre Point
    Edit Point   Subroutines
    Post Point
```

Ready. Messages:0

```
Time.Task.0Update.CRT Dialog
   Input
   Output
   Local
      Local
      Activity
      DetailKey
      Catch
      BaseWorks
      TargetRelations
      ServerSideValidation
      Map
      WsyBase
      Work
         CAS.Area
         PRJ.Project
         SPR.Sub project
         TSK.Status
         TSK.Task name
         TSK.Start date
         TSK.End date
         TSK.Reference
         TSK.Estimate
         TSK.Priority
         TSK.Task type
         EMP.Employee
         Time.Task.#Update
   Calls
   View
```

Clear Values of All Local Fields

- Reset all local fields
- Switch (+++,+)
  - +++Define
  - +If
- Meta loop (+)
  - *+For Each Variable*
  - *+For Each Field*
- Assignment (++)
  - *++Set Empty*



```
Function  _Event
  Copyright 2004 Computer Associates International, Ir
  Seq   Description
  Go Sub  Initialize
  Pre Point
  Edit Point  Execute
  Post Point
  Sub  Initialize
    Pre Point  Start initialize
      Seq   Check for DWA variant selected
      +++Define   Field: FIELDS/+Subroutine
      Pre Point  Pre initialize
        +++Define   Field: WSYBASE/+MetaVaria
        +++Define   Field: FIELDS/+Subroutine
        Pre Point
        Edit Point  Before initializing local an
        Post Point
        +If  Field: FIELDS/+Subroutine
          +For Each Variable  .Local
            +For Each Field
              ++Set Empty
          +For Each Variable  .Output
        Set  Environment<*Returning status> = <
        Set  Environment<*Object> = <*Object.*B
      Edit Point  Pre initialize
      Post Point
Ready.  Messages:0
```

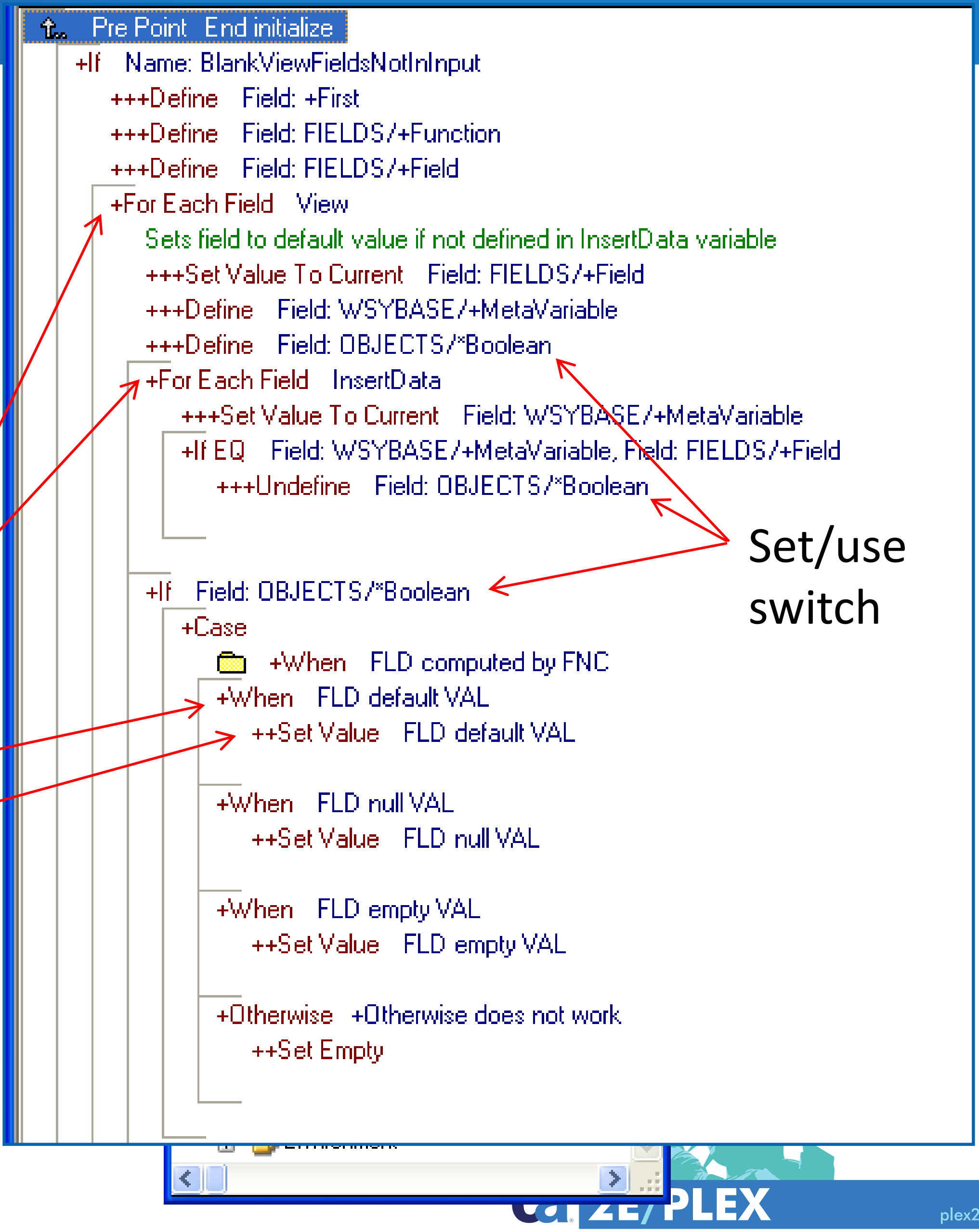
Variable Palette

```
_Event
  Input
  Local
    SignatureFieldSeparator
    Document
    WebInput
    Local
      FieldsToSignature
      SignatureKey
      FieldName
      PropertyValue
      IdentifyViewFields
      WsyCharacter
      DateMask
      FieldLength
      WsyLength
      PositionWsyd
      WsyString
      TimeMask
      TimestampMask
      TokenIsDate
      DecimalSeparator
      ThousandSeparator
      DecimalSeparatorCast
      FieldNumeric
      FormFieldLength
      FormFieldLengthNumeric
      FormFieldSize
      _Work.Num7
    WorkFields
    DateParts
    TimeParts
    WsyLocal 2
    Hidden
    ExcludeFromSign
    MemInputForPageGenerator
    SysWebInput
    TargetRelations
    WsyBase
```

Specific Problems Solved by Meta

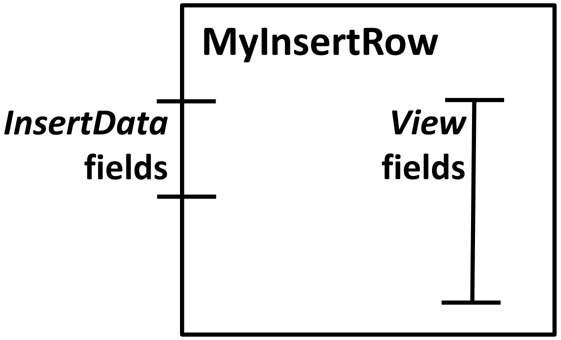
Reset *View* Fields not in *InsertData* Variable

- *View* fields set from *InsertData* variable
  - In other edit point
- Reset *View* fields not in *InsertData* variable
- Remarks
  - Meta loop through fields in variable
  - Examine field triples
  - Code line generated



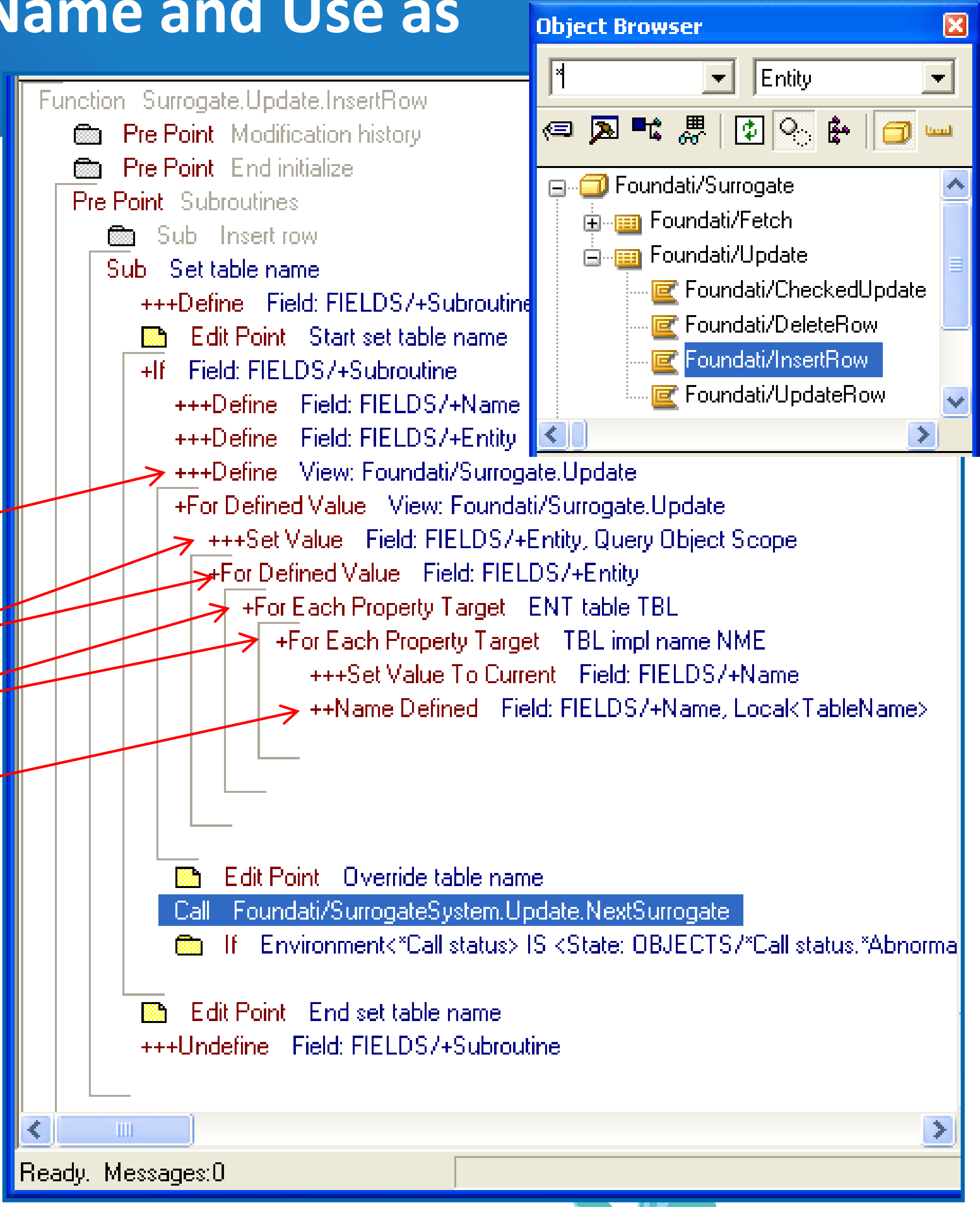
```
Pre Point  End initialize
  +If    Name: BlankViewFieldsNotInInput
    +++Define    Field: +First
    +++Define    Field: FIELDS/+Function
    +++Define    Field: FIELDS/+Field
    +For Each Field    View
      Sets field to default value if not defined in InsertData variable
      +++Set Value To Current    Field: FIELDS/+Field
      +++Define    Field: WSYBASE/+MetaVariable
      +++Define    Field: OBJECTS/*Boolean
      +For Each Field    InsertData
        +++Set Value To Current    Field: WSYBASE/+MetaVariable
        +If EQ    Field: WSYBASE/+MetaVariable, Field: FIELDS/+Field
          +++Undefine    Field: OBJECTS/*Boolean

      +If    Field: OBJECTS/*Boolean
        +Case
            +When    FLD computed by FNC
          +When    FLD default VAL
            ++Set Value    FLD default VAL

          +When    FLD null VAL
            ++Set Value    FLD null VAL

          +When    FLD empty VAL
            ++Set Value    FLD empty VAL

          +Otherwise    +Otherwise does not work
            ++Set Empty
```

Set/use switch



Get Table Implementation Name and Use as Parameter

- Retrieve implementation name of table object
- Remarks
  - Point to view object
  - Step up in scope structure
  - Examine model triples
  - Get value for mapping
    - Only line of generated code



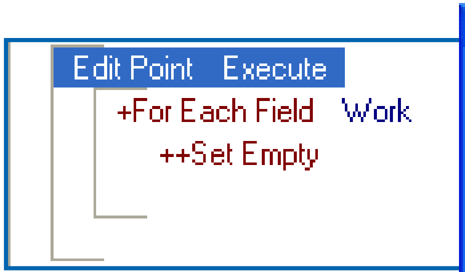
```
Function   Surrogate.Update.InsertRow
    Pre Point   Modification history
    Pre Point   End initialize
  Pre Point   Subroutines
      Sub   Insert row
    Sub   Set table name
      +++Define   Field: FIELDS/+Subroutine
        Edit Point   Start set table name
      +If   Field: FIELDS/+Subroutine
        +++Define   Field: FIELDS/+Name
        +++Define   Field: FIELDS/+Entity
        +++Define   View: Foundati/Surrogate.Update
        +For Defined Value   View: Foundati/Surrogate.Update
          +++Set Value   Field: FIELDS/+Entity, Query Object Scope
          +For Defined Value   Field: FIELDS/+Entity
            +For Each Property Target   ENT table TBL
              +For Each Property Target   TBL impl name NME
                +++Set Value To Current   Field: FIELDS/+Name
                ++Name Defined   Field: FIELDS/+Name, Local<TableName>
        Edit Point   Override table name
        Call   Foundati/SurrogateSystem.Update.NextSurrogate
        If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnorma
      Edit Point   End set table name
      +++Undefine   Field: FIELDS/+Subroutine
```

Ready. Messages:0

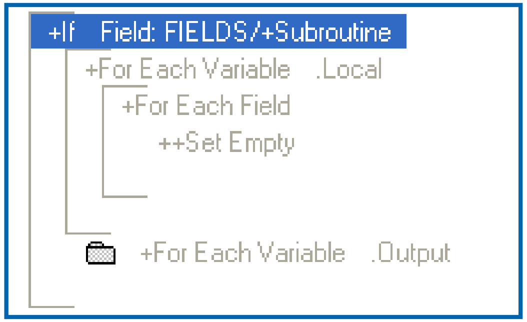
What Can Be Done with Meta?

Meta Loops to Query Contents of Variable Palette

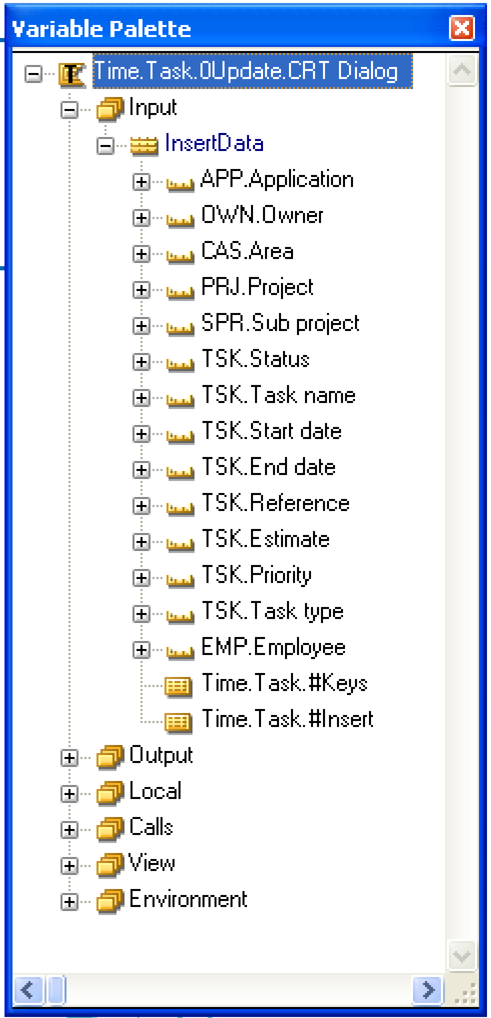
- Meta loops to traverse contents of Variable Palette
  - Loop through all fields in a variable
  - Loop through all variables in ‘group
- A number of loop options
  - Variable
  - View
  - Relations
  - Fields
    - All, view fields, non-view fields, relation attributes
- ++Set empty or further query
  - E.g. inspect model triples



```
Edit Point   Execute
    +For Each Field   Work
        ++Set Empty
```

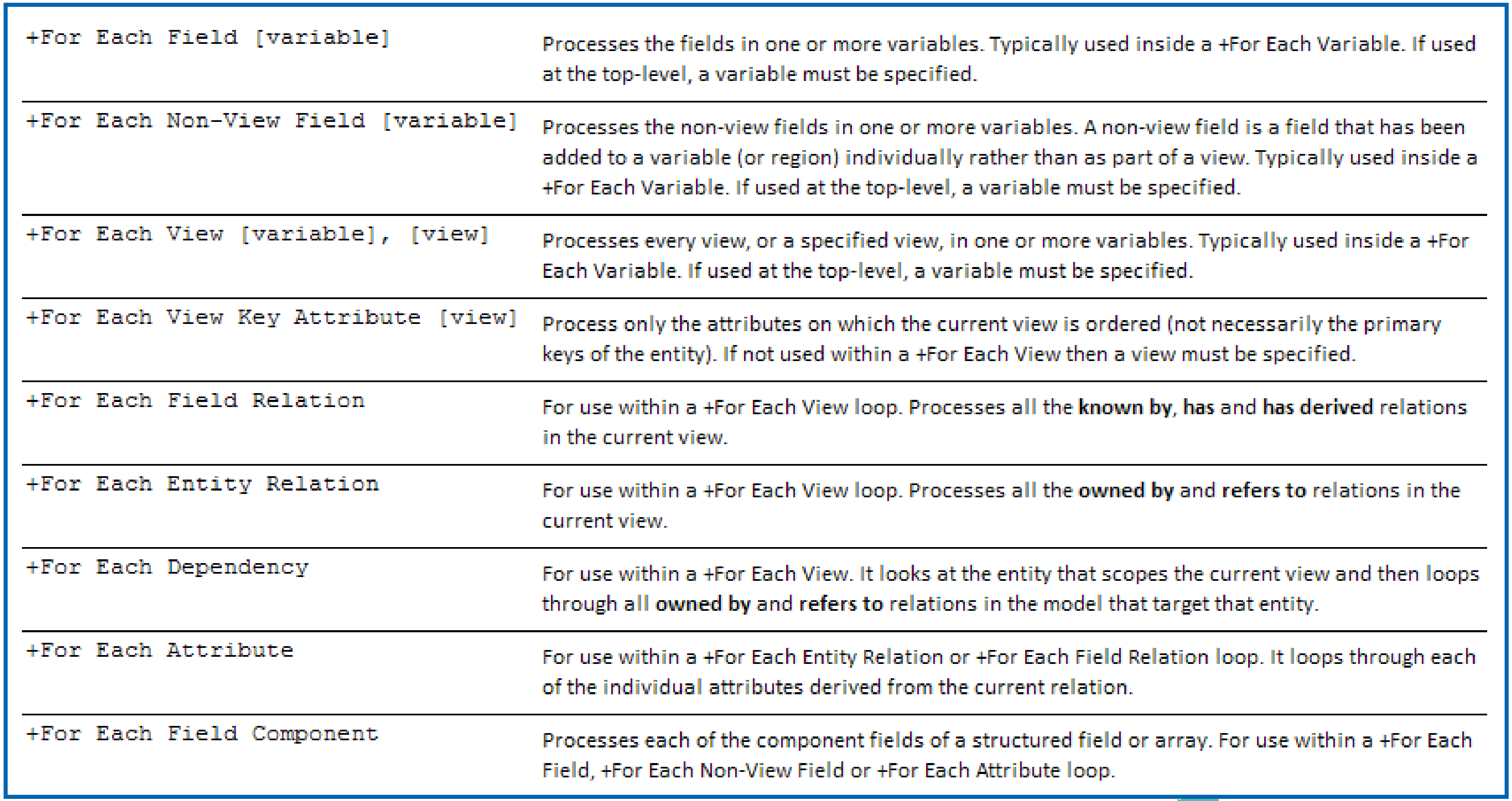


```
+If   Field: FIELDS/+Subroutine
    +For Each Variable   .Local
        +For Each Field
            ++Set Empty

    +For Each Variable   .Output
```



```
Time.Task.0Update.CRT Dialog
    Input
        InsertData
            APP.Application
            OWN.Owner
            CAS.Area
            PRJ.Project
            SPR.Sub project
            TSK.Status
            TSK.Task name
            TSK.Start date
            TSK.End date
            TSK.Reference
            TSK.Estimate
            TSK.Priority
            TSK.Task type
            EMP.Employee
            Time.Task.#Keys
            Time.Task.#Insert
    Output
    Local
    Calls
    View
    Environment
```

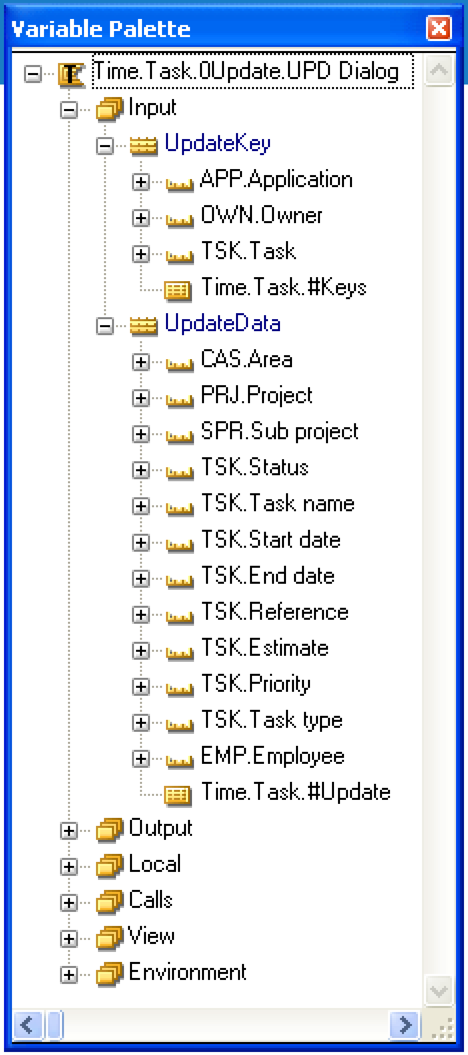
Meta Loops to Query Contents of Variable Palette



`+For Each Field [variable]`	Processes the fields in one or more variables. Typically used inside a +For Each Variable. If used at the top-level, a variable must be specified.
`+For Each Non-View Field [variable]`	Processes the non-view fields in one or more variables. A non-view field is a field that has been added to a variable (or region) individually rather than as part of a view. Typically used inside a +For Each Variable. If used at the top-level, a variable must be specified.
`+For Each View [variable], [view]`	Processes every view, or a specified view, in one or more variables. Typically used inside a +For Each Variable. If used at the top-level, a variable must be specified.
`+For Each View Key Attribute [view]`	Process only the attributes on which the current view is ordered (not necessarily the primary keys of the entity). If not used within a +For Each View then a view must be specified.
`+For Each Field Relation`	For use within a +For Each View loop. Processes all the **known by**, **has** and **has derived** relations in the current view.
`+For Each Entity Relation`	For use within a +For Each View loop. Processes all the **owned by** and **refers to** relations in the current view.
`+For Each Dependency`	For use within a +For Each View. It looks at the entity that scopes the current view and then loops through all **owned by** and **refers to** relations in the model that target that entity.
`+For Each Attribute`	For use within a +For Each Entity Relation or +For Each Field Relation loop. It loops through each of the individual attributes derived from the current relation.
`+For Each Field Component`	Processes each of the component fields of a structured field or array. For use within a +For Each Field, +For Each Non-View Field or +For Each Attribute loop.

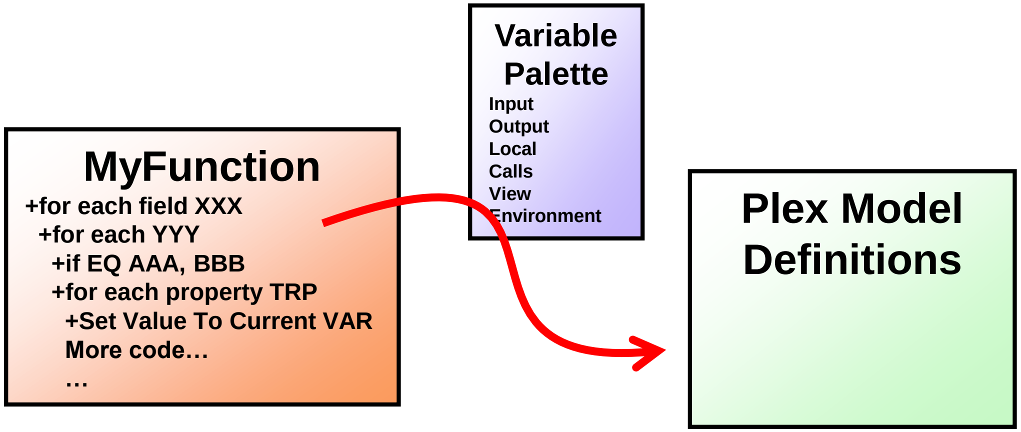
Meta Loops to Query Contents of Variable Palette

- Inspect model definitions (triples) by Meta
  - Inspect triples directly
  - Inspect Variable Palette
- Resolving fields in variable is non-trivial
  - Specific and inherited
  - View fields (contains, omit)
  - Replacements
- Current function -> Variable Palette -> Model



Variable Palette

```
⊟ Time.Task.0Update.UPD Dialog
  ⊟ Input
    ⊟ UpdateKey
      ⊞ APP.Application
      ⊞ OWN.Owner
      ⊞ TSK.Task
        Time.Task.#Keys
    ⊟ UpdateData
      ⊞ CAS.Area
      ⊞ PRJ.Project
      ⊞ SPR.Sub project
      ⊞ TSK.Status
      ⊞ TSK.Task name
      ⊞ TSK.Start date
      ⊞ TSK.End date
      ⊞ TSK.Reference
      ⊞ TSK.Estimate
      ⊞ TSK.Priority
      ⊞ TSK.Task type
      ⊞ EMP.Employee
        Time.Task.#Update
  ⊞ Output
  ⊞ Local
  ⊞ Calls
  ⊞ View
  ⊞ Environment
```



MyFunction

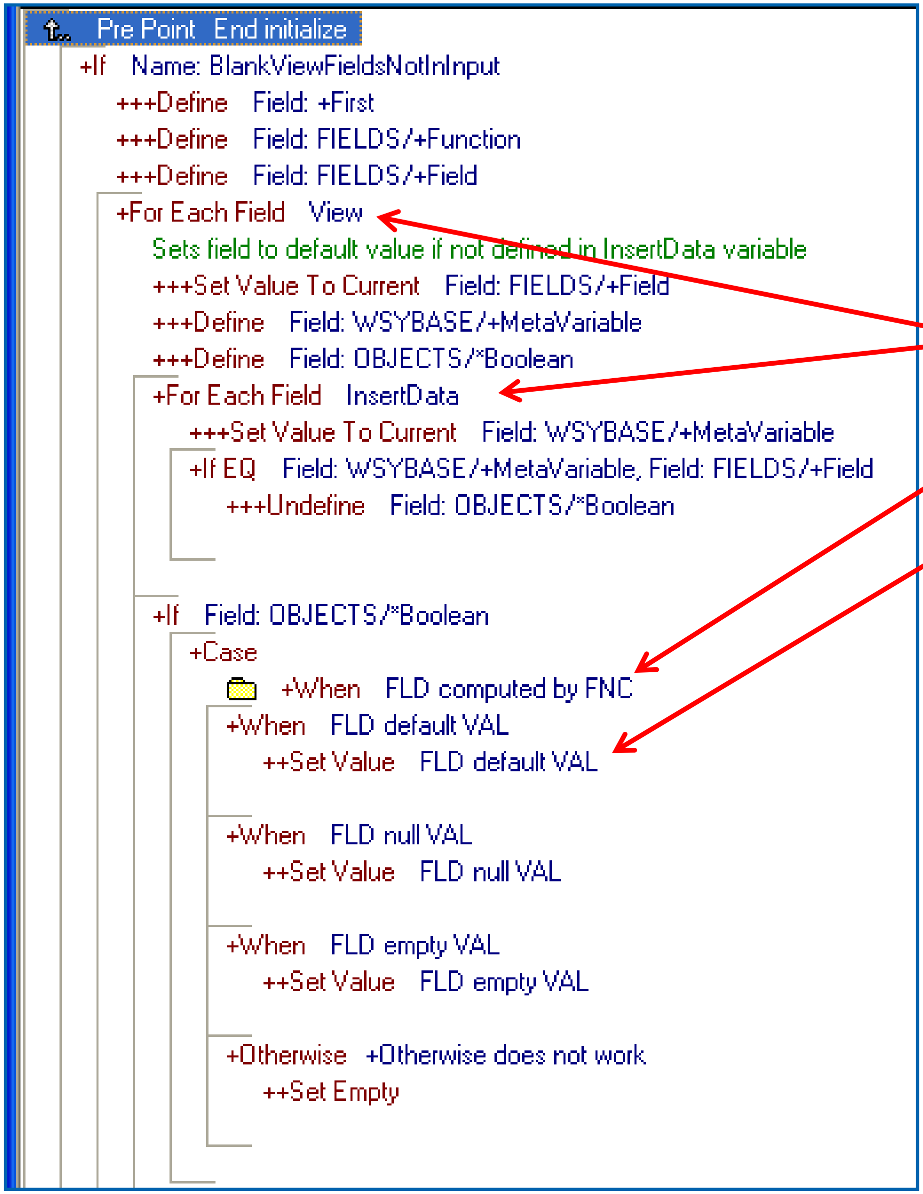
```
+for each field XXX
  +for each YYY
    +if EQ AAA, BBB
    +for each property TRP
      +Set Value To Current VAR
      More code…
      …
```

Variable Palette

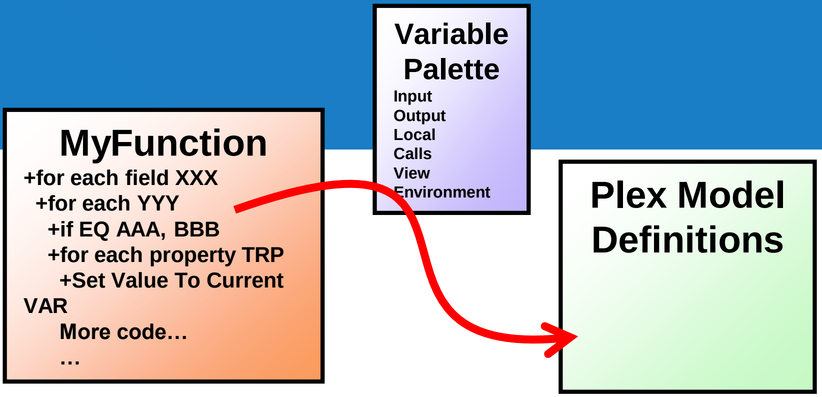
Input
Output
Local
Calls
View
Environment

Plex Model Definitions

Query Model Definitions (Triples)



```
Pre Point   End initialize
  +If   Name: BlankViewFieldsNotInInput
    +++Define   Field: +First
    +++Define   Field: FIELDS/+Function
    +++Define   Field: FIELDS/+Field
    +For Each Field   View
      Sets field to default value if not defined in InsertData variable
      +++Set Value To Current   Field: FIELDS/+Field
      +++Define   Field: WSYBASE/+MetaVariable
      +++Define   Field: OBJECTS/*Boolean
      +For Each Field   InsertData
        +++Set Value To Current   Field: WSYBASE/+MetaVariable
        +If EQ   Field: WSYBASE/+MetaVariable, Field: FIELDS/+Field
          +++Undefine   Field: OBJECTS/*Boolean

      +If   Field: OBJECTS/*Boolean
        +Case
            +When   FLD computed by FNC
          +When   FLD default VAL
            ++Set Value   FLD default VAL

          +When   FLD null VAL
            ++Set Value   FLD null VAL

          +When   FLD empty VAL
            ++Set Value   FLD empty VAL

          +Otherwise   +Otherwise does not work
            ++Set Empty
```



Variable Palette

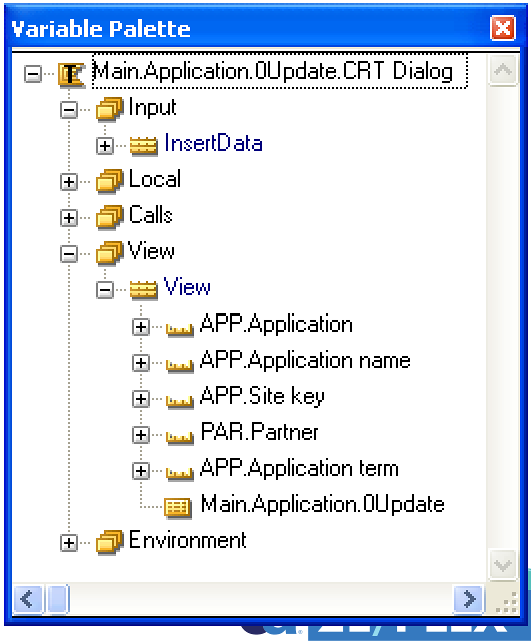
- Input
- Output
- Local
- Calls
- View
- Environment

MyFunction

```
+for each field XXX
  +for each YYY
    +if EQ AAA, BBB
    +for each property TRP
      +Set Value To Current
VAR
      More code…
      …
```

Plex Model Definitions

- Start from Variable Palette
  - Query model triples
  - Generate code

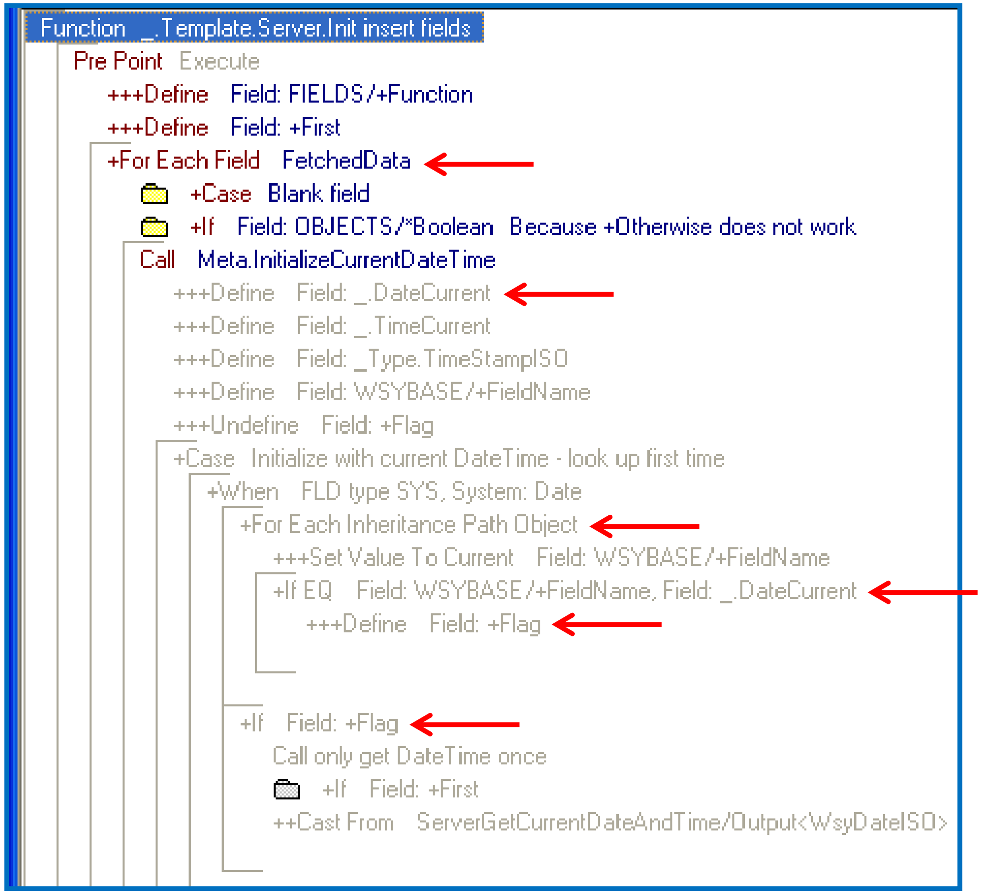


```
⊟ Main.Application.0Update.CRT Dialog
   ⊟ Input
      ⊞ InsertData
   ⊞ Local
   ⊞ Calls
   ⊟ View
      ⊟ View
         ⊞ APP.Application
         ⊞ APP.Application name
         ⊞ APP.Site key
         ⊞ PAR.Partner
         ⊞ APP.Application term
           Main.Application.0Update
   ⊞ Environment
```

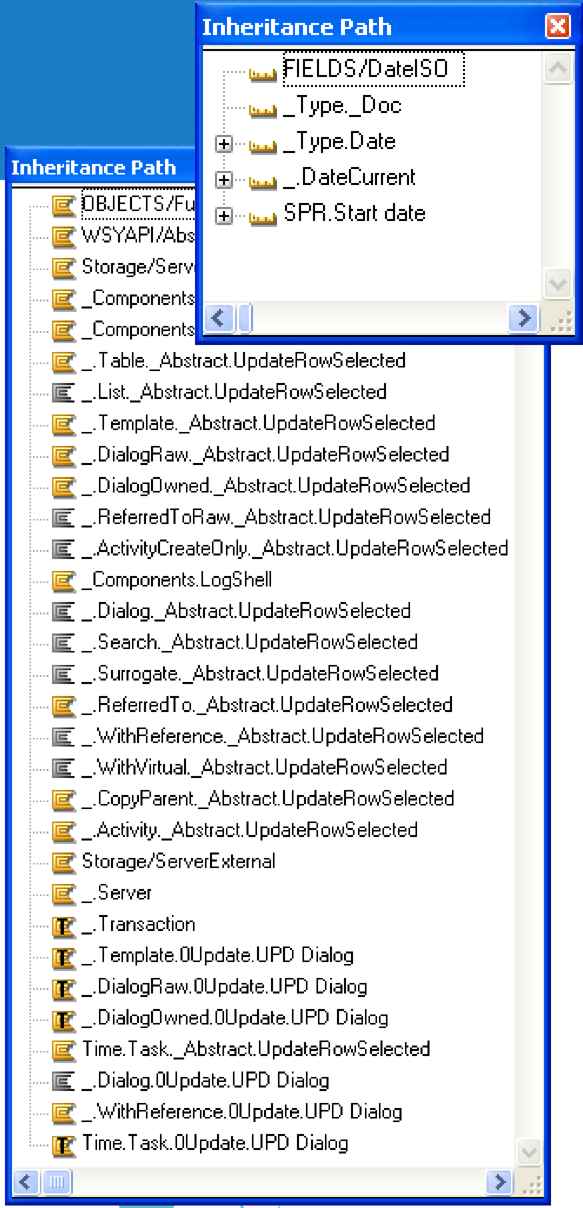
Loop/Query Inheritance Path

- Meta pseudo code:

```
+For specified model object
    +If Specified ObjectX in inheritance part
        Generate portion of code
```



```
Function   _.Template.Server.Init insert fields
  Pre Point   Execute
    +++Define   Field: FIELDS/+Function
    +++Define   Field: +First
    +For Each Field   FetchedData
        +Case   Blank field
        +If   Field: OBJECTS/*Boolean   Because +Otherwise does not work
      Call   Meta.InitializeCurrentDateTime
        +++Define   Field: _.DateCurrent
        +++Define   Field: _.TimeCurrent
        +++Define   Field: _Type.TimeStampISO
        +++Define   Field: WSYBASE/+FieldName
        +++Undefine   Field: +Flag
        +Case   Initialize with current DateTime - look up first time
          +When   FLD type SYS, System: Date
            +For Each Inheritance Path Object
              +++Set Value To Current   Field: WSYBASE/+FieldName
              +If EQ   Field: WSYBASE/+FieldName, Field: _.DateCurrent
                +++Define   Field: +Flag

            +If   Field: +Flag
              Call only get DateTime once
              +If   Field: +First
              ++Cast From   ServerGetCurrentDateAndTime/Output<WsyDateISO>
```



Inheritance Path

- FIELDS/DateISO
- _Type._Doc
- _Type.Date
- _.DateCurrent
- SPR.Start date

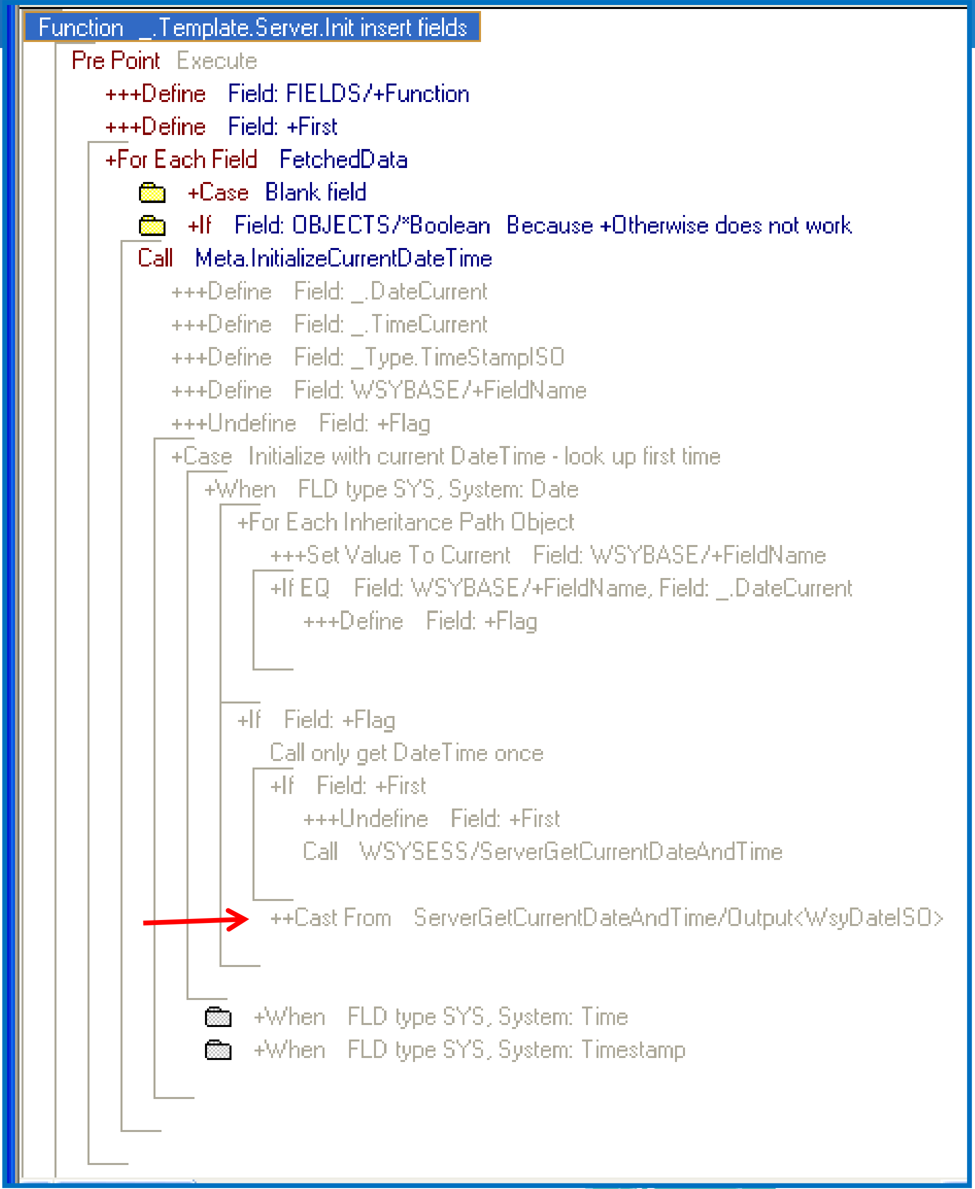
Inheritance Path

- OBJECTS/Fu
- WSYAPI/Abs
- Storage/Serv
- _Components
- _Components
- _.Table._Abstract.UpdateRowSelected
- _.List._Abstract.UpdateRowSelected
- _.Template._Abstract.UpdateRowSelected
- _.DialogRaw._Abstract.UpdateRowSelected
- _.DialogOwned._Abstract.UpdateRowSelected
- _.ReferredToRaw._Abstract.UpdateRowSelected
- _.ActivityCreateOnly._Abstract.UpdateRowSelected
- _Components.LogShell
- _.Dialog._Abstract.UpdateRowSelected
- _.Search._Abstract.UpdateRowSelected
- _.Surrogate._Abstract.UpdateRowSelected
- _.ReferredTo._Abstract.UpdateRowSelected
- _.WithReference._Abstract.UpdateRowSelected
- _.WithVirtual._Abstract.UpdateRowSelected
- _.CopyParent._Abstract.UpdateRowSelected
- _.Activity._Abstract.UpdateRowSelected
- Storage/ServerExternal
- _.Server
- _.Transaction
- _.Template.0Update.UPD Dialog
- _.DialogRaw.0Update.UPD Dialog
- _.DialogOwned.0Update.UPD Dialog
- Time.Task._Abstract.UpdateRowSelected
- _.Dialog.0Update.UPD Dialog
- _.WithReference.0Update.UPD Dialog
- Time.Task.0Update.UPD Dialog

Getting and Setting Values from Meta Loops

- Statement: *++Cast To*
  - Cast value of Field pointed to by Meta to specified Variable Palette field
- Statement: *++Cast From*

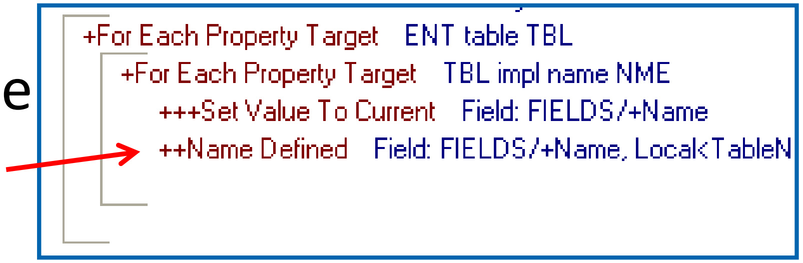
  Cast value of specified Variable Palette field to Field pointed to by Meta



```
Function  _.Template.Server.Init insert fields
   Pre Point  Execute
      +++Define   Field: FIELDS/+Function
      +++Define   Field: +First
      +For Each Field   FetchedData
         +Case  Blank field
         +If   Field: OBJECTS/*Boolean  Because +Otherwise does not work
         Call   Meta.InitializeCurrentDateTime
            +++Define   Field: _.DateCurrent
            +++Define   Field: _.TimeCurrent
            +++Define   Field: _Type.TimeStampISO
            +++Define   Field: WSYBASE/+FieldName
            +++Undefine   Field: +Flag
            +Case  Initialize with current DateTime - look up first time
               +When   FLD type SYS, System: Date
                  +For Each Inheritance Path Object
                     +++Set Value To Current   Field: WSYBASE/+FieldName
                     +If EQ   Field: WSYBASE/+FieldName, Field: _.DateCurrent
                        +++Define   Field: +Flag

                  +If   Field: +Flag
                     Call only get DateTime once
                     +If   Field: +First
                        +++Undefine   Field: +First
                        Call   WSYSESS/ServerGetCurrentDateAndTime

                     ++Cast From   ServerGetCurrentDateAndTime/Output<WsyDateISO>


               +When   FLD type SYS, System: Time
               +When   FLD type SYS, System: Timestamp
```

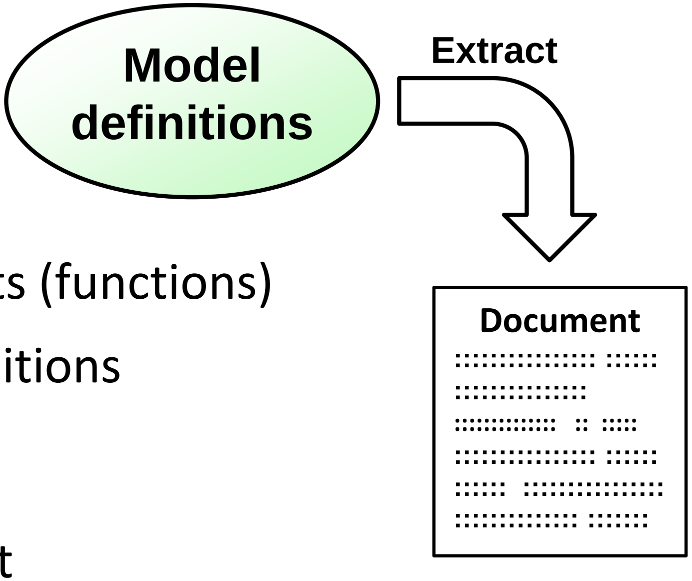
Extracting Documentation/Definitions from Model

- Extract and use implementation name as part of code



```
+For Each Property Target   ENT table TBL
  +For Each Property Target   TBL impl name NME
    +++Set Value To Current   Field: FIELDS/+Name
    ++Name Defined   Field: FIELDS/+Name, Local<TableN
```

- Statements to retrieve model definitions
  - ++Name Defined
  - ++Text Defined

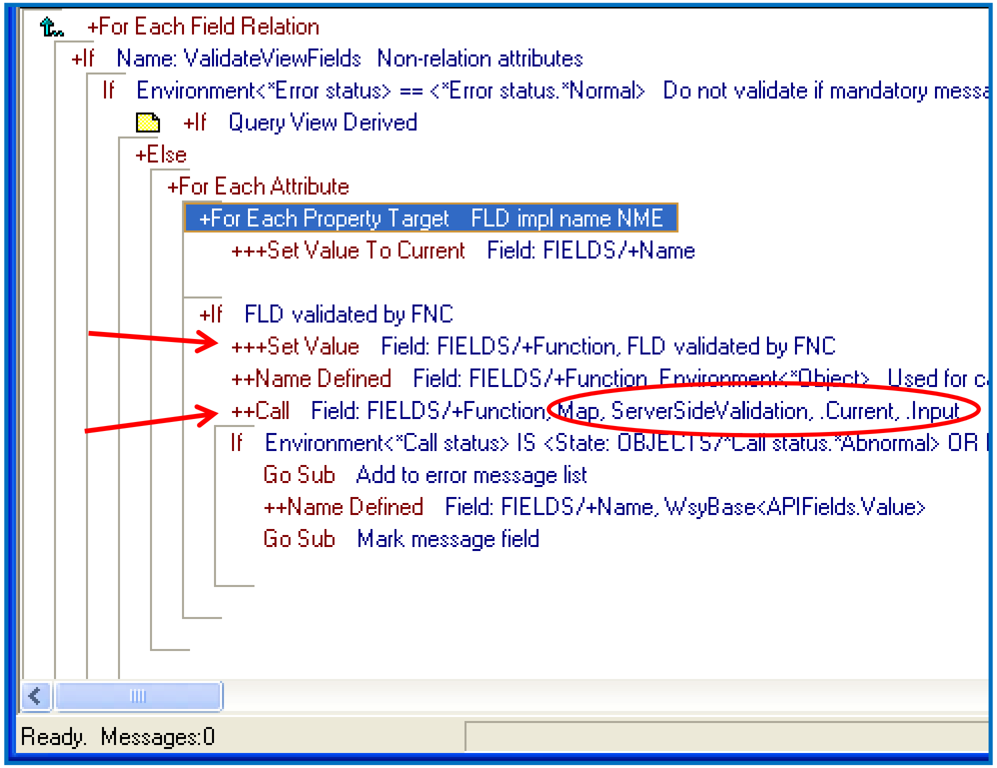


- Websydian template generators
  - Extract definitions from multiple model objects (functions)
  - Generate template documents based on definitions
- Extract model documentation
  - Extract specific model definitions to document
  - Validate design rules…
  - Consider Plex API as an alternative

Meta Calls

MyField	Includes FNC	Validate
	validated by FNC	Validate

- If ‘rule triple’ is specified
  - Call function
- Call comprised function
  - *FNC comprises FNC*
- Specify default parameter mapping
  - Specific mapping not possible
- Returned parameters must be specified as Dual
  - Function called by Meta will not appear in Variable Palette



```
+For Each Field Relation
 +If   Name: ValidateViewFields  Non-relation attributes
   If   Environment<*Error status> == <*Error status.*Normal>   Do not validate if mandatory messa
      +If   Query View Derived
    +Else
      +For Each Attribute
        +For Each Property Target   FLD impl name NME
          +++Set Value To Current   Field: FIELDS/+Name

        +If   FLD validated by FNC
          +++Set Value   Field: FIELDS/+Function, FLD validated by FNC
          ++Name Defined   Field: FIELDS/+Function, Environment<*Object>   Used for c
          ++Call   Field: FIELDS/+Function, Map, ServerSideValidation, .Current, .Input
          If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnormal> OR I
            Go Sub   Add to error message list
            ++Name Defined   Field: FIELDS/+Name, WsyBase<APIFields.Value>
            Go Sub   Mark message field
```

Ready. Messages:0

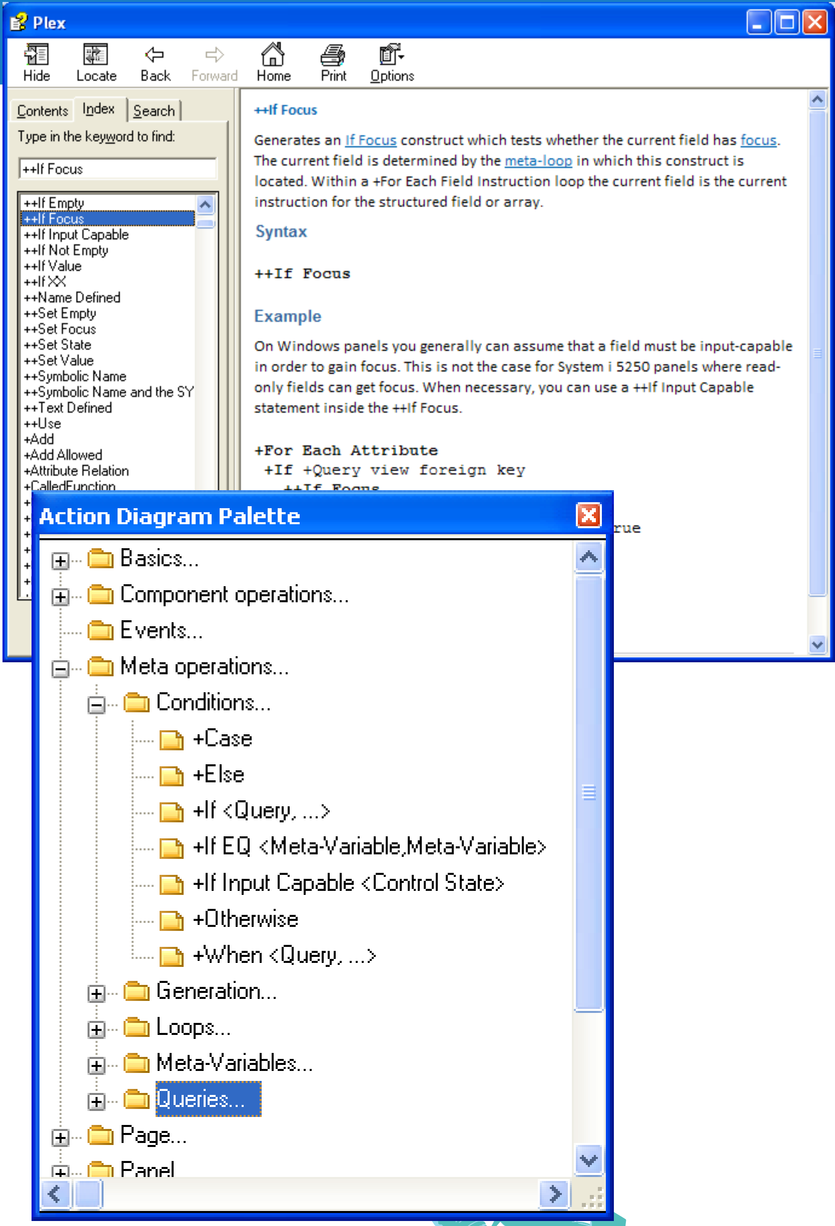
Working with Meta Coding

Getting Comfortable with Plex Meta Code

- Investigate and understand existing implementations and get comfortable with these, e.g.
  - *WSYBASE/PageGenerator* and *WSYBASE/EventHandler*
  - *FOUNDATION/Surrogate.Update.InsertRow*
  - *UIBASIC/Update*
- Conduct your own experiments
- Trial and error
  - Syntax may be difficult, but debugging Meta is easy…
- Study the Plex help

Examples and Sources of Information

- Plex Help
  - Context, Specifics, and Examples
- Action Diagram Palette
  - Contains entire syntax
- Examples
  - Websydian Pages, Events and Template Generators
  - UI functions
  - Fetch and Update functions
  - Own implementations



++If Focus

Generates an If Focus construct which tests whether the current field has focus. The current field is determined by the meta-loop in which this construct is located. Within a +For Each Field Instruction loop the current field is the current instruction for the structured field or array.

Syntax

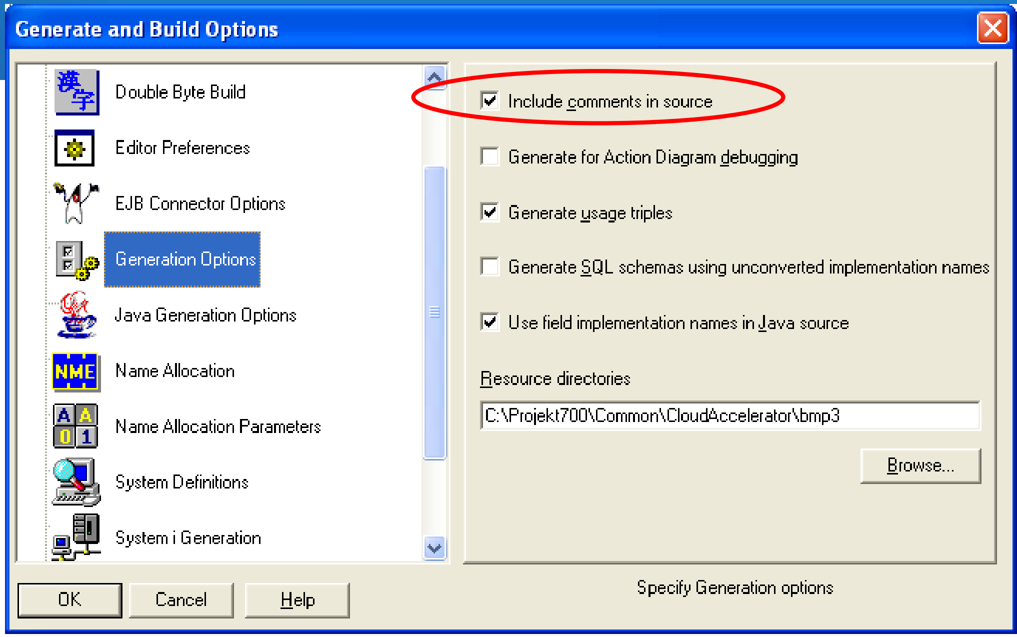
```
++If Focus
```

Example

On Windows panels you generally can assume that a field must be input-capable in order to gain focus. This is not the case for System i 5250 panels where read-only fields can get focus. When necessary, you can use a ++If Input Capable statement inside the ++If Focus.

```
+For Each Attribute
 +If +Query view foreign key
  ++If Focus
```

Debugging Meta by Investigating Generated Source



☑ Include comments in source

□ Generate for Action Diagram debugging

☑ Generate usage triples

□ Generate SQL schemas using unconverted implementation names

☑ Use field implementation names in Java source

Resource directories

C:\Projekt700\Common\CloudAccelerator\bmp3

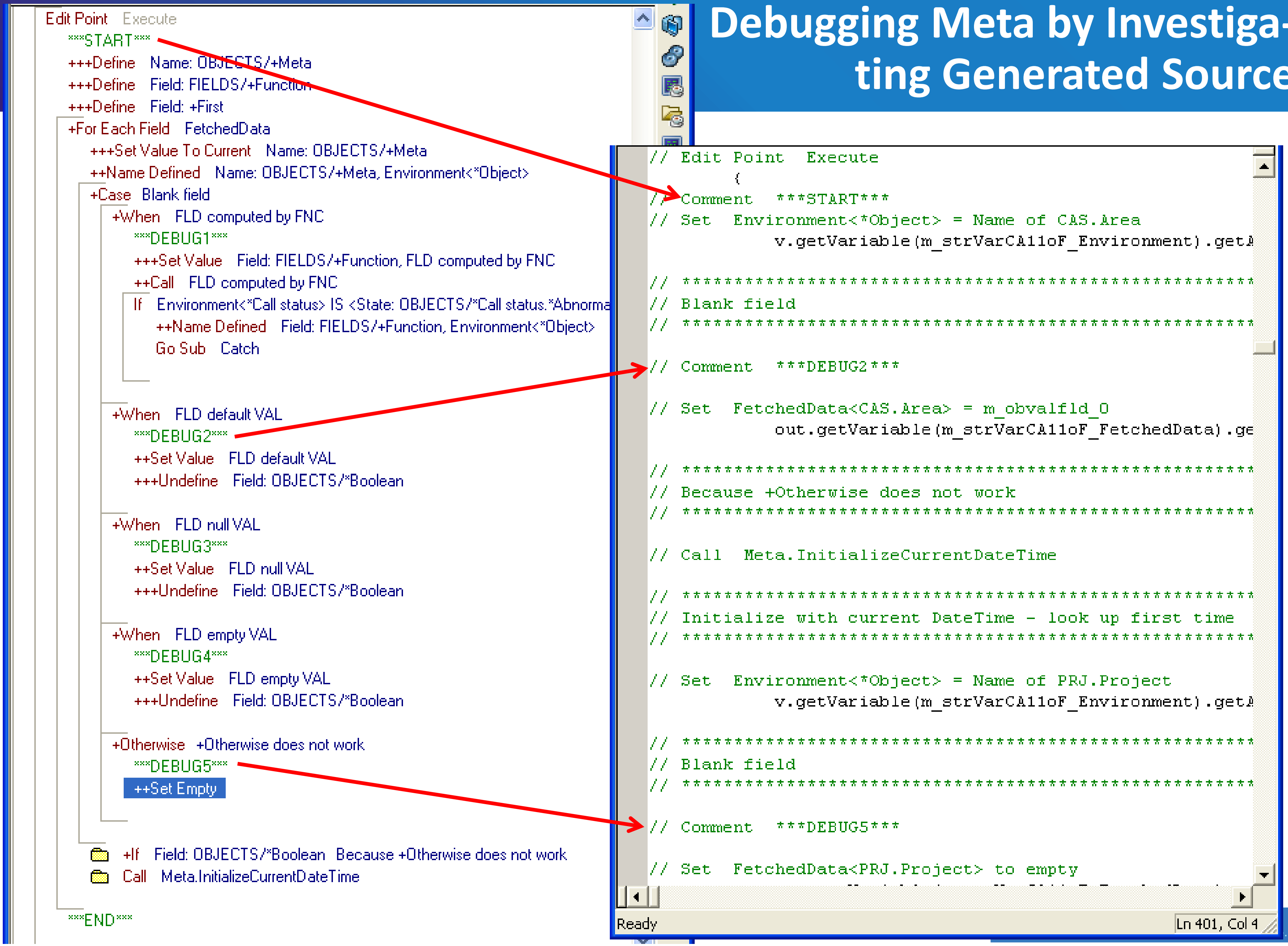
Specify Generation options

- Generate and Build Options
  - Set *Include comments in source*
- Insert comment markers
  - Code portion generated or not
  - *Comment ***START HERE****
- Investigate Meta pointers/values by inserting code

  +++Define

  ++Name Defined
- ‘Debug code’ may be inserted at abstract level
  - Or entire abstract portion copied and debugged upon…

Debugging Meta by Investigating Generated Source



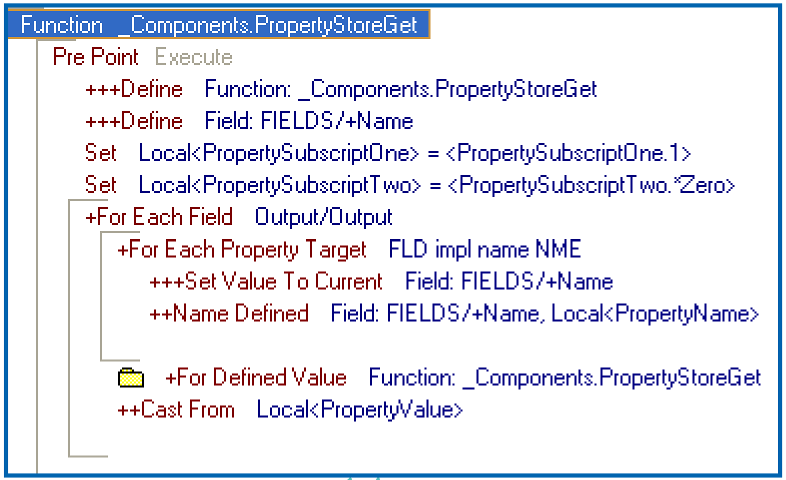
```
Edit Point   Execute
  ***START***
  +++Define   Name: OBJECTS/+Meta
  +++Define   Field: FIELDS/+Function
  +++Define   Field: +First
  +For Each Field   FetchedData
    +++Set Value To Current   Name: OBJECTS/+Meta
    ++Name Defined   Name: OBJECTS/+Meta, Environment<*Object>
    +Case   Blank field
      +When   FLD computed by FNC
        ***DEBUG1***
        +++Set Value   Field: FIELDS/+Function, FLD computed by FNC
        ++Call   FLD computed by FNC
        If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnorma
          ++Name Defined   Field: FIELDS/+Function, Environment<*Object>
          Go Sub   Catch

      +When   FLD default VAL
        ***DEBUG2***
        ++Set Value   FLD default VAL
        +++Undefine   Field: OBJECTS/*Boolean

      +When   FLD null VAL
        ***DEBUG3***
        ++Set Value   FLD null VAL
        +++Undefine   Field: OBJECTS/*Boolean

      +When   FLD empty VAL
        ***DEBUG4***
        ++Set Value   FLD empty VAL
        +++Undefine   Field: OBJECTS/*Boolean

      +Otherwise   +Otherwise does not work
        ***DEBUG5***
        ++Set Empty

    +If   Field: OBJECTS/*Boolean   Because +Otherwise does not work
    Call   Meta.InitializeCurrentDateTime
  ***END***
```

```
// Edit Point   Execute
       {
// Comment   ***START***
// Set   Environment<*Object> = Name of CAS.Area
            v.getVariable(m_strVarCAllof_Environment).getA

// ***********************************************************
// Blank field
// ***********************************************************

// Comment   ***DEBUG2***

// Set   FetchedData<CAS.Area> = m_obvalfld_0
            out.getVariable(m_strVarCAllof_FetchedData).ge

// ***********************************************************
// Because +Otherwise does not work
// ***********************************************************

// Call   Meta.InitializeCurrentDateTime

// ***********************************************************
// Initialize with current DateTime - look up first time
// ***********************************************************

// Set   Environment<*Object> = Name of PRJ.Project
            v.getVariable(m_strVarCAllof_Environment).getA

// ***********************************************************
// Blank field
// ***********************************************************

// Comment   ***DEBUG5***

// Set   FetchedData<PRJ.Project> to empty
```

Ready   Ln 401, Col 4

Understand Meta Variables, Meta loops, and Meta Pointers 1

- Any object in Plex model can be used as a meta variable
  - Make code readable by use specific objects, e.g. *FIELDS/+Field*, *FIELDS/+Function*
  - All Meta variables must be defined before use
    - Or no code is generated (typical error)
- Meta loops
  - Variable Palette loops
  - Inheritance Path loop
  - Statement: *+For Each Property [Target]*
    - Investigate model triples
    - One or more loop iterations
    - May be called within Variable Palette loops
    - Nested loops within loops

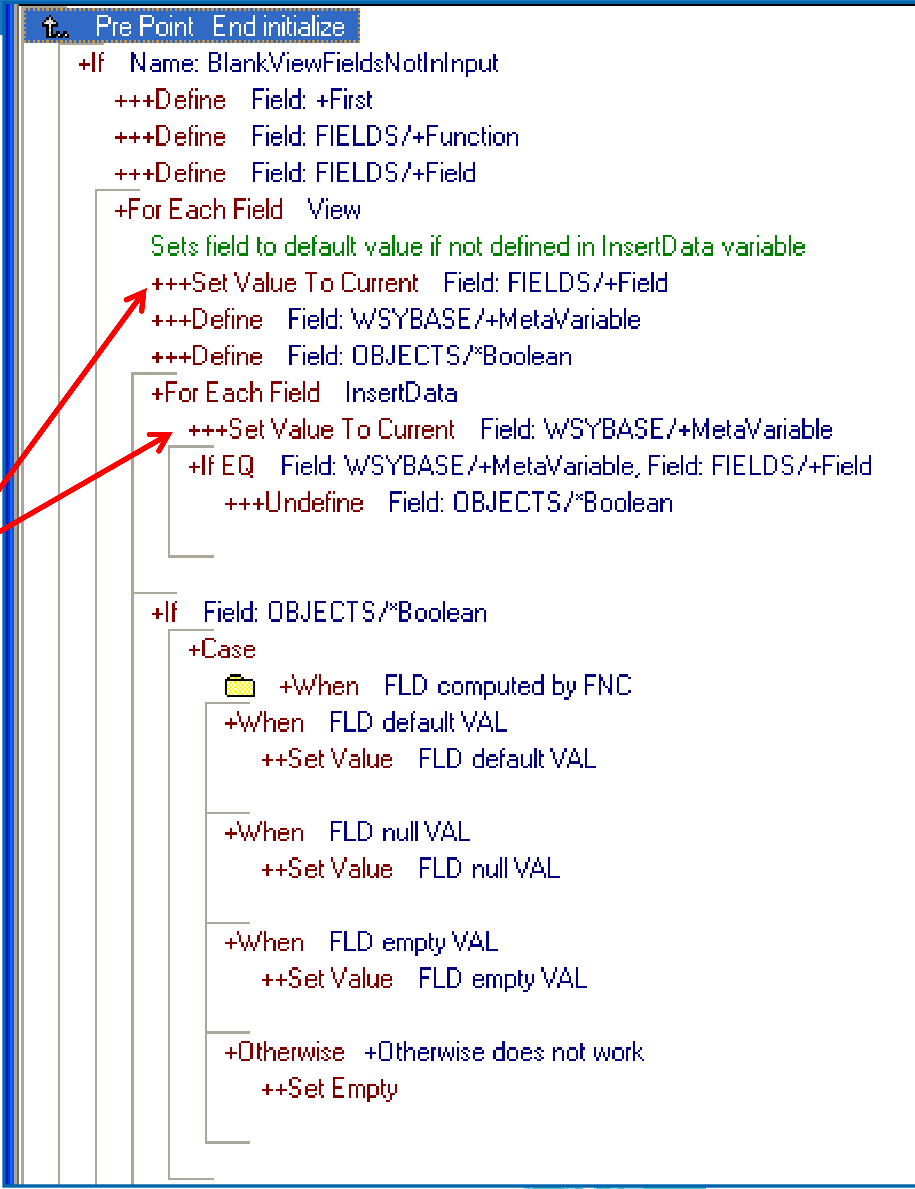


```
Function   _Components.PropertyStoreGet
  Pre Point  Execute
    +++Define   Function: _Components.PropertyStoreGet
    +++Define   Field: FIELDS/+Name
    Set   Local<PropertySubscriptOne> = <PropertySubscriptOne.1>
    Set   Local<PropertySubscriptTwo> = <PropertySubscriptTwo.*Zero>
    +For Each Field   Output/Output
      +For Each Property Target   FLD impl name NME
        +++Set Value To Current   Field: FIELDS/+Name
        ++Name Defined   Field: FIELDS/+Name, Local<PropertyName>

        +For Defined Value   Function: _Components.PropertyStoreGet
      ++Cast From   Local<PropertyValue>
```

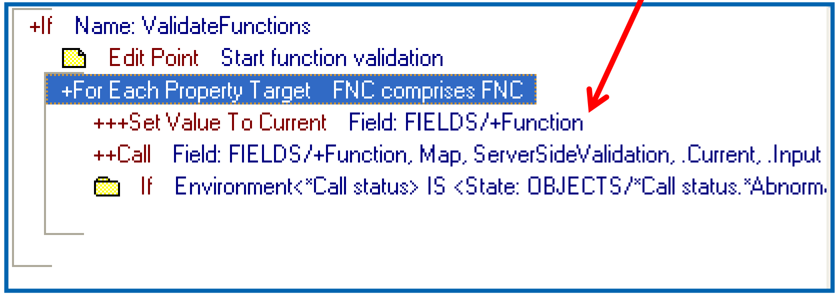
Understand Meta Variables, Meta loops, and Meta Pointers 2

- ‘Current Pointer’ within meta loop
  - Target?
- Save value of Current Pointer

  +++Set Value To Current MetaVar



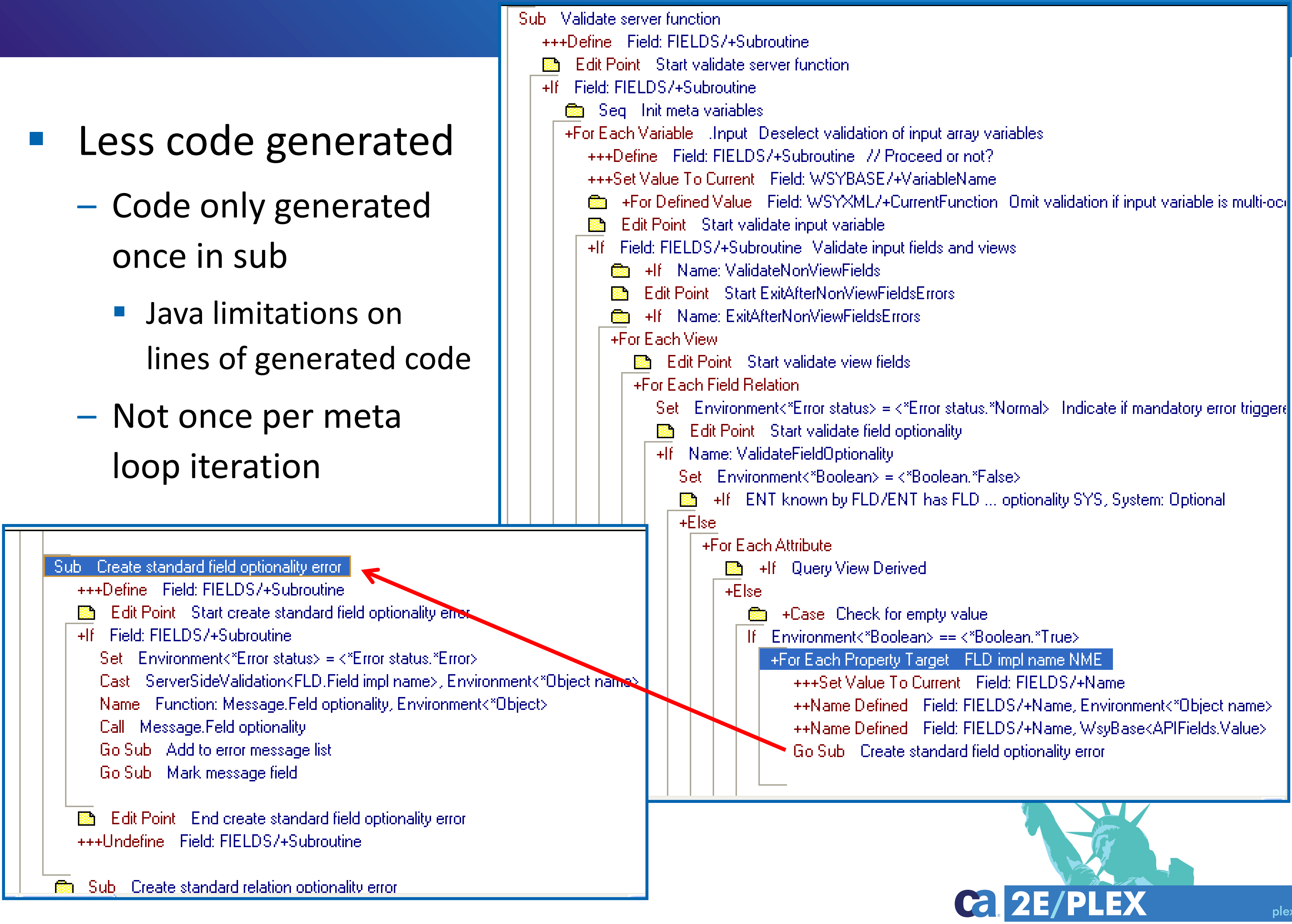
```
Pre Point  End initialize
  +If   Name: BlankViewFieldsNotInInput
    +++Define   Field: +First
    +++Define   Field: FIELDS/+Function
    +++Define   Field: FIELDS/+Field
    +For Each Field   View
      Sets field to default value if not defined in InsertData variable
      +++Set Value To Current   Field: FIELDS/+Field
      +++Define   Field: WSYBASE/+MetaVariable
      +++Define   Field: OBJECTS/*Boolean
      +For Each Field   InsertData
        +++Set Value To Current   Field: WSYBASE/+MetaVariable
        +If EQ   Field: WSYBASE/+MetaVariable, Field: FIELDS/+Field
          +++Undefine   Field: OBJECTS/*Boolean

      +If   Field: OBJECTS/*Boolean
        +Case
            +When   FLD computed by FNC
          +When   FLD default VAL
            ++Set Value   FLD default VAL

          +When   FLD null VAL
            ++Set Value   FLD null VAL

          +When   FLD empty VAL
            ++Set Value   FLD empty VAL

          +Otherwise   +Otherwise does not work
            ++Set Empty
```



```
+If   Name: ValidateFunctions
    Edit Point   Start function validation
  +For Each Property Target   FNC comprises FNC
    +++Set Value To Current   Field: FIELDS/+Function
    ++Call   Field: FIELDS/+Function, Map, ServerSideValidation, .Current, .Input
      If   Environment<*Call status> IS <State: OBJECTS/*Call status.*Abnorm.
```

Delegate ‘Static’ Code to Sub

- Less code generated
  - Code only generated once in sub
    - Java limitations on lines of generated code
  - Not once per meta loop iteration

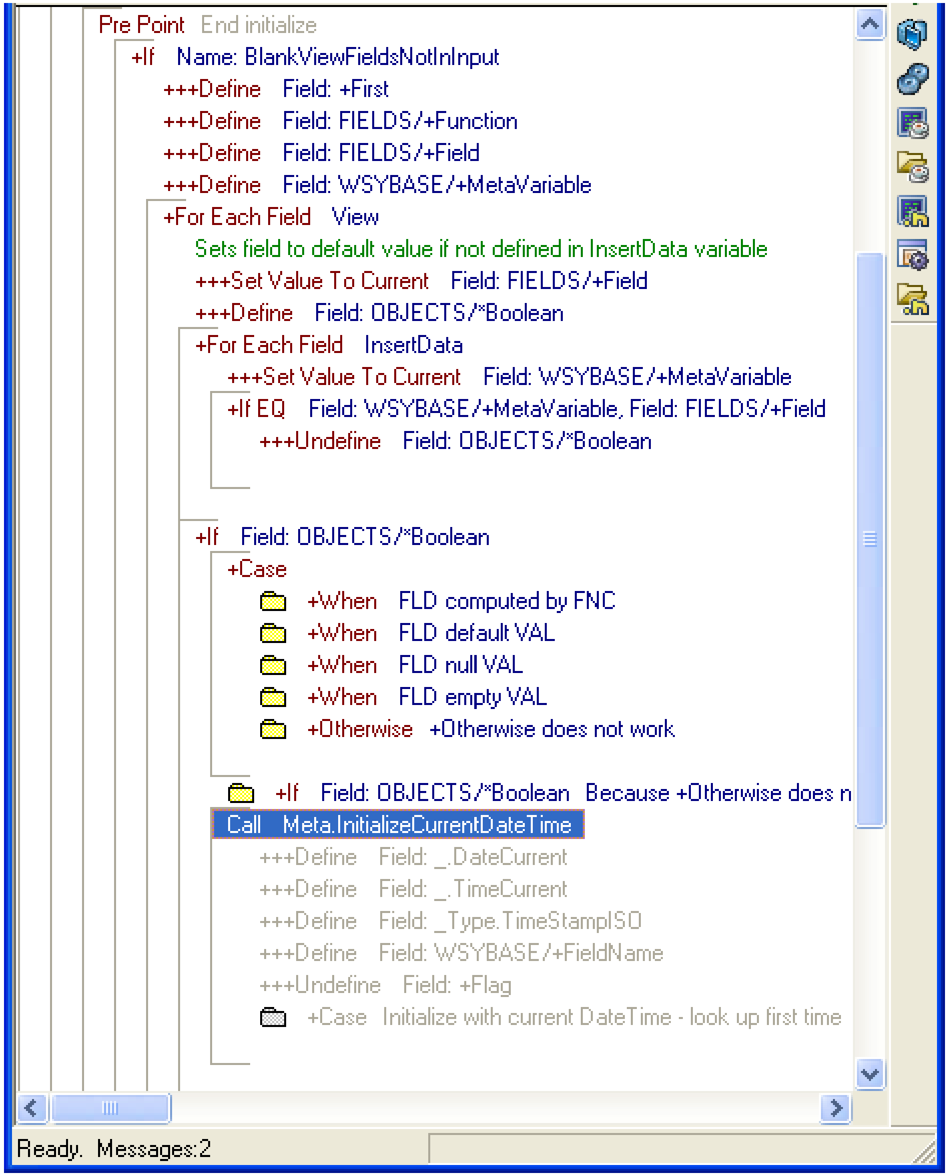


```
Sub  Validate server function
  +++Define  Field: FIELDS/+Subroutine
  Edit Point  Start validate server function
  +If  Field: FIELDS/+Subroutine
    Seq  Init meta variables
    +For Each Variable  .Input  Deselect validation of input array variables
      +++Define  Field: FIELDS/+Subroutine  // Proceed or not?
      +++Set Value To Current  Field: WSYBASE/+VariableName
      +For Defined Value  Field: WSYXML/+CurrentFunction  Omit validation if input variable is multi-oc
      Edit Point  Start validate input variable
      +If  Field: FIELDS/+Subroutine  Validate input fields and views
        +If  Name: ValidateNonViewFields
        Edit Point  Start ExitAfterNonViewFieldsErrors
        +If  Name: ExitAfterNonViewFieldsErrors
        +For Each View
          Edit Point  Start validate view fields
          +For Each Field Relation
            Set  Environment<*Error status> = <*Error status.*Normal>  Indicate if mandatory error triggere
            Edit Point  Start validate field optionality
            +If  Name: ValidateFieldOptionality
              Set  Environment<*Boolean> = <*Boolean.*False>
              +If  ENT known by FLD/ENT has FLD ... optionality SYS, System: Optional
              +Else
                +For Each Attribute
                  +If  Query View Derived
                  +Else
                    +Case  Check for empty value
                    If  Environment<*Boolean> == <*Boolean.*True>
                      +For Each Property Target  FLD impl name NME
                        +++Set Value To Current  Field: FIELDS/+Name
                        ++Name Defined  Field: FIELDS/+Name, Environment<*Object name>
                        ++Name Defined  Field: FIELDS/+Name, WsyBase<APIFields.Value>
                        Go Sub  Create standard field optionality error
```

```
Sub  Create standard field optionality error
  +++Define  Field: FIELDS/+Subroutine
  Edit Point  Start create standard field optionality error
  +If  Field: FIELDS/+Subroutine
    Set  Environment<*Error status> = <*Error status.*Error>
    Cast  ServerSideValidation<FLD.Field impl name>, Environment<*Object name>
    Name  Function: Message.Feld optionality, Environment<*Object>
    Call  Message.Feld optionality
    Go Sub  Add to error message list
    Go Sub  Mark message field

  Edit Point  End create standard field optionality error
  +++Undefine  Field: FIELDS/+Subroutine
Sub  Create standard relation optionality error
```

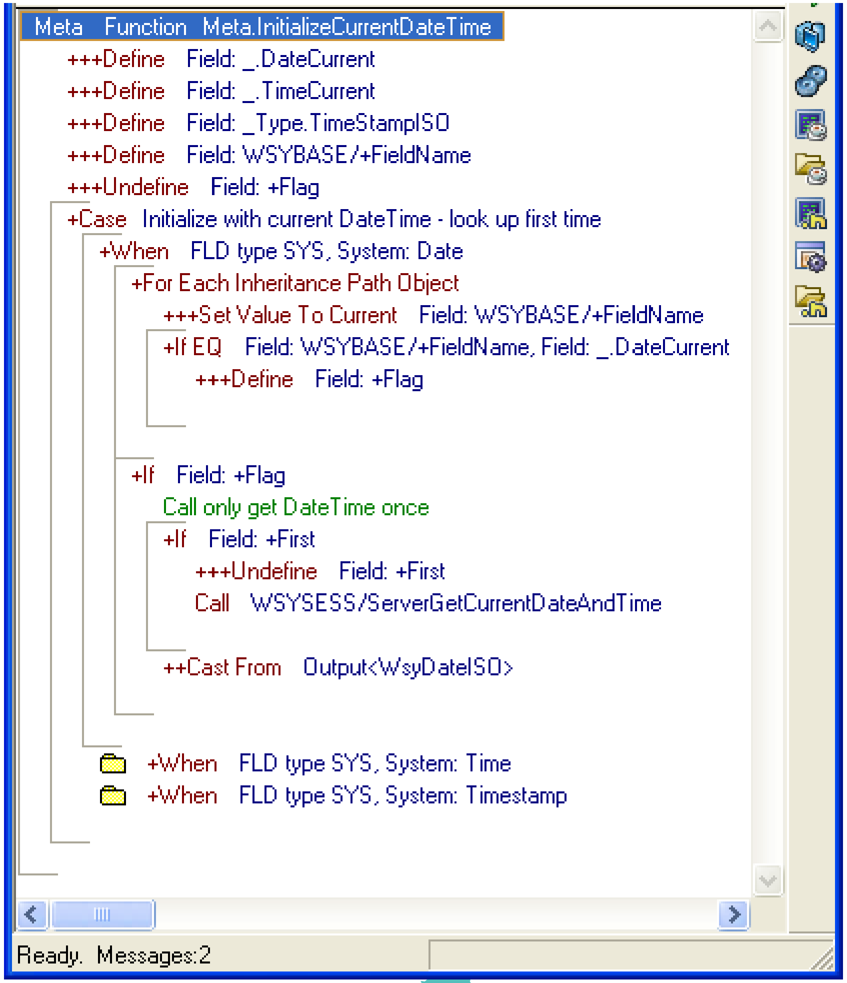
ca 2E/PLEX plex2e.com
2013 WORLDWIDE DEVELOPER CONFERENCE

Specify Reusable Code in Meta Functions



```
Pre Point   End initialize
  +If   Name: BlankViewFieldsNotInInput
    +++Define   Field: +First
    +++Define   Field: FIELDS/+Function
    +++Define   Field: FIELDS/+Field
    +++Define   Field: WSYBASE/+MetaVariable
    +For Each Field   View
      Sets field to default value if not defined in InsertData variable
      +++Set Value To Current   Field: FIELDS/+Field
      +++Define   Field: OBJECTS/*Boolean
      +For Each Field   InsertData
        +++Set Value To Current   Field: WSYBASE/+MetaVariable
        +If EQ   Field: WSYBASE/+MetaVariable, Field: FIELDS/+Field
          +++Undefine   Field: OBJECTS/*Boolean

      +If   Field: OBJECTS/*Boolean
        +Case
          +When   FLD computed by FNC
          +When   FLD default VAL
          +When   FLD null VAL
          +When   FLD empty VAL
          +Otherwise   +Otherwise does not work

        +If   Field: OBJECTS/*Boolean   Because +Otherwise does n
        Call   Meta.InitializeCurrentDateTime
          +++Define   Field: _.DateCurrent
          +++Define   Field: _.TimeCurrent
          +++Define   Field: _Type.TimeStampISO
          +++Define   Field: WSYBASE/+FieldName
          +++Undefine   Field: +Flag
          +Case   Initialize with current DateTime - look up first time
```

Ready. Messages:2

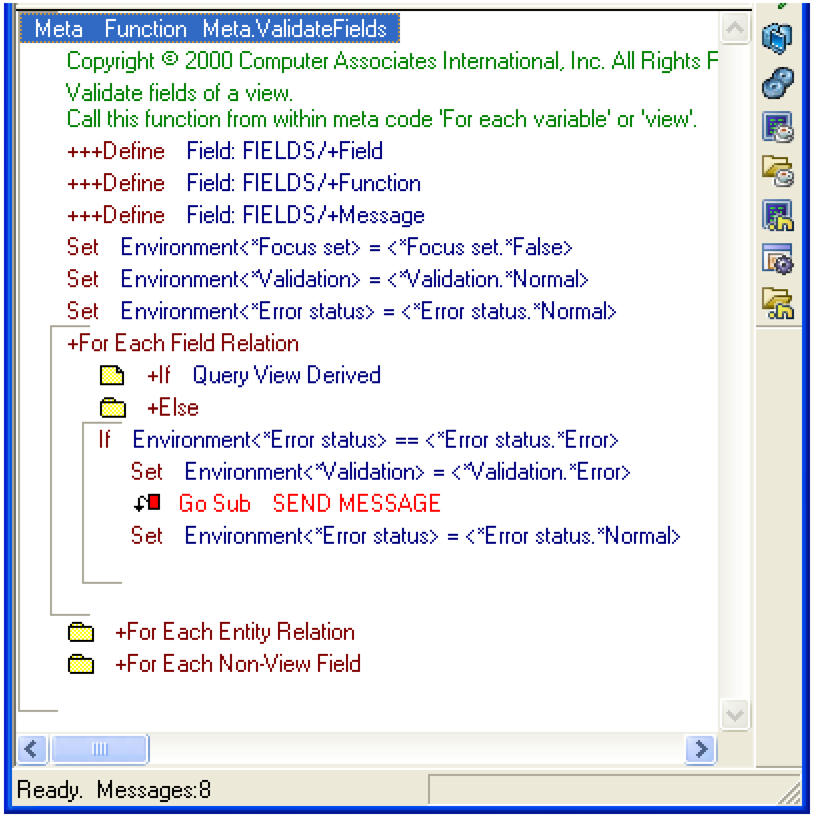


```
Meta   Function   Meta.InitializeCurrentDateTime
  +++Define   Field: _.DateCurrent
  +++Define   Field: _.TimeCurrent
  +++Define   Field: _Type.TimeStampISO
  +++Define   Field: WSYBASE/+FieldName
  +++Undefine   Field: +Flag
  +Case   Initialize with current DateTime - look up first time
    +When   FLD type SYS, System: Date
      +For Each Inheritance Path Object
        +++Set Value To Current   Field: WSYBASE/+FieldName
        +If EQ   Field: WSYBASE/+FieldName, Field: _.DateCurrent
          +++Define   Field: +Flag

      +If   Field: +Flag
        Call only get DateTime once
        +If   Field: +First
          +++Undefine   Field: +First
          Call   WSYSESS/ServerGetCurrentDateAndTime

        ++Cast From   Output<WsyDateISO>

    +When   FLD type SYS, System: Time
    +When   FLD type SYS, System: Timestamp
```

Ready. Messages:2

Specify Reusable Code in Meta Functions - Rules

- Function type: Meta
- Variable Palette fields must be specified in calling function
- No Edit points
- Go Sub statements can be defined
  - Even though target Sub is not specified yet



```
Meta  Function  Meta.ValidateFields
    Copyright © 2000 Computer Associates International, Inc. All Rights F
    Validate fields of a view.
    Call this function from within meta code 'For each variable' or 'view'.
    +++Define   Field: FIELDS/+Field
    +++Define   Field: FIELDS/+Function
    +++Define   Field: FIELDS/+Message
    Set   Environment<*Focus set> = <*Focus set.*False>
    Set   Environment<*Validation> = <*Validation.*Normal>
    Set   Environment<*Error status> = <*Error status.*Normal>
    +For Each Field Relation
        +If   Query View Derived
        +Else
      If   Environment<*Error status> == <*Error status.*Error>
          Set   Environment<*Validation> = <*Validation.*Error>
          Go Sub   SEND MESSAGE
          Set   Environment<*Error status> = <*Error status.*Normal>
    +For Each Entity Relation
    +For Each Non-View Field
Ready.  Messages:8
```

Q&A