

IBM Enterpak (Application Development Solutions) TeamConnection Online Help

IBM Education and Training, Atlanta GA / Raleigh NC
May - September 1996

TeamConnection is an **IBM VisualAge** tool that provides full life cycle development for an application. It is an object-based, open information model using an ObjectStore database in a TCP/IP network client/server environment with build support for OS2 and MVS platforms.

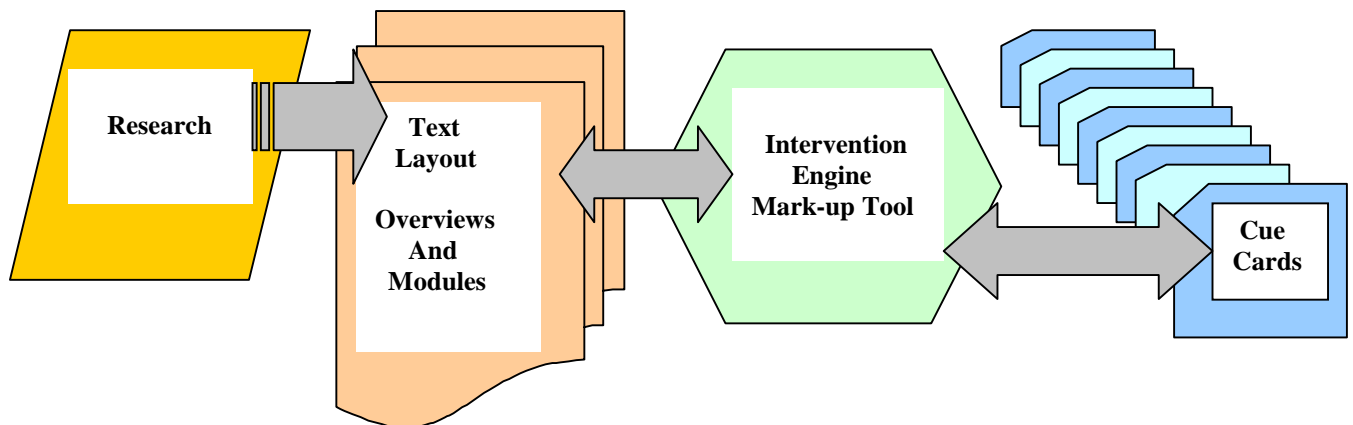
Though a useful and necessary tool for application development, TeamConnection does require that a programmer learn and adhere to certain conventions beyond that of simply coding in COBOL or C++. For this reason **IBM Enterpak** elected to develop performance support help in the form of on-line cue cards, to be accessed on the fly as the programmer works.

The project ran from May through September 1996, at the IBM Education and Training office in Atlanta, GA. The team included a project manager, an instructional designer, a lead technical writer (my role), a junior writer, a writer intern, a programmer intern, the mark-up tool team, and SMEs from IBM's Raleigh technology lab.

Completed help text content included a TeamConnection **Process Overview**, each application's **Overview**, and step-by-step "**Performance Assistant**" Modules. Content research was collected from manuals, seminar notes, online databases, and subject matter expert (SME) interviews. Text was developed using Lotus WordPro, transferred to the intervention engine and compiled. The final output resulted in over 1500 cue cards.

The following abstracts from the project include examples of:

- Text content for overviews and modules
- Cue card images
- The intervention engine code



TeamConnection Process Overview

Below are abstracts from the TeamConnection *Process Overview* in text format, and the resulting cue cards.

Card 80000 (Introduction to Process Overview)

Welcome to the Process Overview What the Overview does

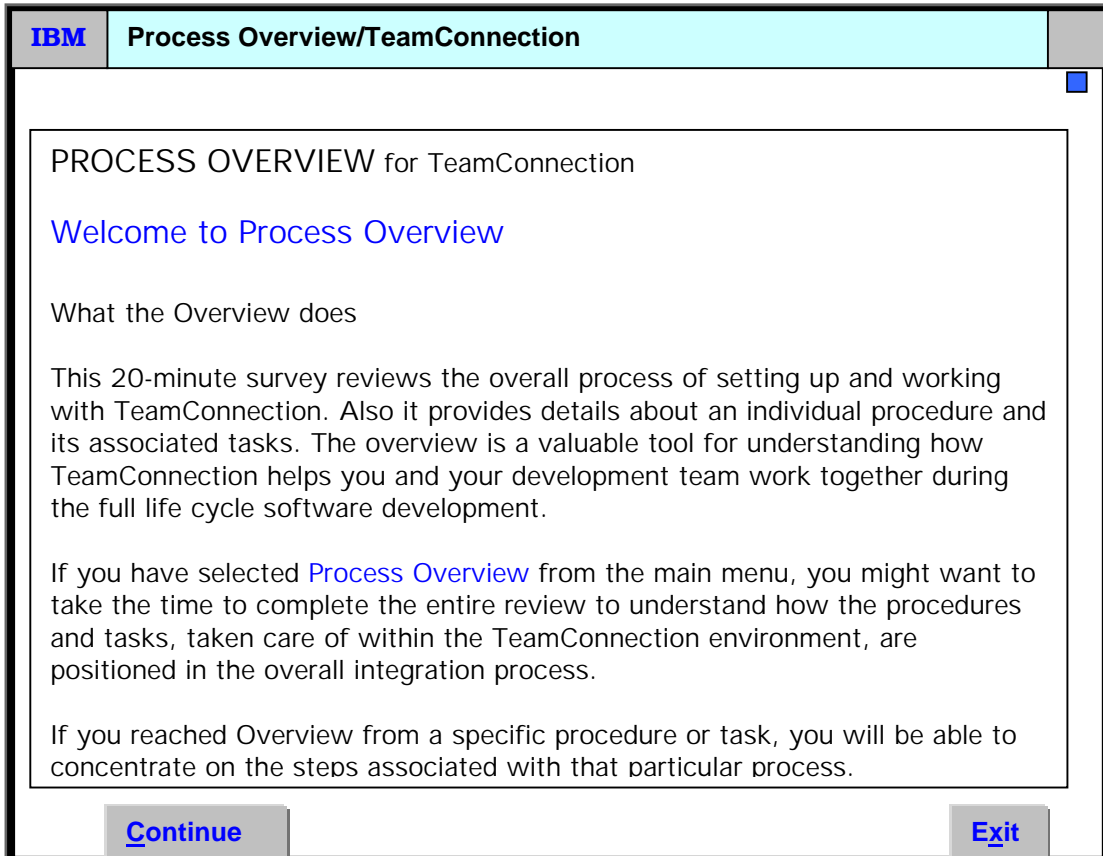
This 20-minute survey reviews the overall process of setting up and working with TeamConnection. Also it provides details about an individual procedure and its associated tasks. The overview is a valuable tool for understanding how TeamConnection helps you and your development team work together during the full life cycle software development.

If you have selected [Process Overview](#) from the main menu, you might want to take the time to complete the entire review to understand how the procedures and tasks, taken care of within the TeamConnection environment, are positioned in the overall integration process.

If you reached Overview from a specific procedure or task, you will be able to concentrate on the steps associated with that particular process.

* * * * *

Cue Card Example



* * * * *

Card 81000 (MAIN MENU Process Overview)

Overview Topics Menu

Overview is written as a logical sequence of text on a series of cue cards, introducing TeamConnection concepts and providing a description of procedures and tasks in the order they would be performed by lead programmers and the programmers under them.

Navigation buttons are provided at the bottom of each cue card. Blue text is used for emphasis only, and cannot be clicked to "jump" to that word or concept.

Make your selection from the topics below.

TeamConnection Concepts
Prepare the TeamConnection Environment
Work with Users
Work with Components
Work with Releases
Use Work Areas
Work with Parts
Work with Defects
Work with Features
Work with Drivers
Work with Records
Build an Application
Test an Application
Package an Application
Distribute an Application
Miscellaneous Performance Tips

* * * * *

Cue Card Example

IBM Process Overview/TeamConnection

Overview Topics Menu

Overview is written as a logical sequence of text on a series of cue cards, introducing TeamConnection concepts and providing a description of procedures and tasks in the order they would be performed by lead programmers and the programmers under them.

Navigation buttons are provided at the bottom of each cue card. Blue text is used for emphasis only, and cannot be clicked to "jump" to that word or concept.

Make your selection from the topics below.

- TeamConnection Concepts
- Prepare the TeamConnection Environment
- Work with Users
- Work with Components
- Work with Releases
- Use Work Areas
- Work with Parts
- Work with Defects

OK Previous Exit

Overview for a Specific Application

Overview example for the COBOL application

Card 183400 (COBOL Overview Procedures25)

Prepare the TeamConnection Environment

Your administrator and management team have already configured the TeamConnection Workframe bridge environment for your use. When you begin using TeamConnection with [VisualAge COBOL](#), you'll be able to start with the tasks that relate directly to the development work you do. Configuration was done according to your organization's methods of doing business. If you have any questions about the way something works, you should ask your administrator.

Your installed hardware environment is client/server technology using TCP/IP communications protocol (with a sendmail program), an ObjectStore database, and a sophisticated build facility.

The scope of development work you do under TeamConnection control is driven by the way the TeamConnection elements: family, components, releases and processes have been defined by your organization. This includes data access given to you and your team members.

Your family super user grants initial access by assigning you a userid and including it on a host list of client machines that use TeamConnection. Also, you'll be added to other lists that pertain to authority levels (access list) and message receipts (notification list).

TeamConnection enables Workframe projects for existing [VisualAge COBOL](#) applications are given option values for family, release, work area, queries, and a build rule profile so that the project will be viewed as a set of TeamConnection parts. A tools setup selection makes it possible for you to perform certain TeamConnection part actions directly from the Workframe project.

Performance Assistant Step-by-Step Module

Selections from the *step-by-step module* for VisualAge Generator, the TeamConnection tool to generate and build a completed application.

The *X* in front of a card title was an indication to the intervention engine compiler that a change had been made in that card since the last help compile.

Performance Assistant Main Menu

Select a procedure from the following list to complete one or more of its related tasks. Click the [List All Tasks](#) push button to choose a task from the complete list of tasks.

If you are a new TeamConnection user, click the [Process Overview](#) push button for a description of how TeamConnection works. The Overview takes 20 minutes to complete.

- 1. Prepare the TeamConnection Environment**
- 2. Set Up the TeamConnection Client and MSL TeamConnection Association**
- 3. Prepare Data for Development and Build**
- 4. Complete Development Work**
- 5. Build and Test an Application**
- 6. Package and Distribute an Application**

Ongoing Tasks - Remove and Recover TeamConnection Data

Ongoing Tasks - Modify TeamConnection MetaData

Ongoing Tasks - Run a Query Report

* * * * *

Card 10000 (Procedure1)

Prepare the TeamConnection Environment

As a general user (such as programmer, developer, writer), the TeamConnection environment should be set up for you. This set up includes:

- Establishing the LAN environment and communications system.
- Creating the family.
- Creating components.
- Establishing component ownership.
- Setting up access and notification for all users.
- Creating releases.

This setup also includes turning on certain TeamConnection processes that affect data control and work flow. These processes will affect the work you do. If you have any questions concerning any part of the setup, contact your TeamConnection administrator or project leader, or refer to the *TeamConnection User's Guide*

* * * * *

X Card 20000 (Procedure2 menu)

Set Up the TeamConnection Client and MSL TeamConnection Association

After the TeamConnection Client software is installed on your workstation, you can customize your TeamConnection personal environment. The TeamConnection Client includes the GUI, the command line interface and an interface used by other tools that access objects in the TeamConnection database. The tasks below help you navigate the TeamConnection GUI and issue commands more effectively. Also, you will create Library MSLs associated with TeamConnection by definition. Members, files and outputs are stored as parts to use for your complete development cycle under TeamConnection control.

Double-click a task to read about that topic.

<p>Modify Settings Notebook Edit CONFIG.SYS Add Tasks to Your Task List Create a Library MSL Overview of TeamConnection Naming Conventions</p>

* * * * *

Card 21000 (Procedure2 Task1)

Modify Settings Notebook

Use the Settings Notebook to:

- Set default command attributes.
- Set the relative directory.
- Set the default text editor.
- Customize how the GUI operates.

Consult [Help](#) at the bottom of the each Settings Notebook page in TeamConnection for details, on completing fields.

To modify the Settings Notebook:

1. From the Tasks window, select [Window: Settings](#).
2. Complete the Environment Page. Minimally, specify a default family name, user ID, and relative directory. Click [Apply](#).
3. Complete the Set Up page. Change the [edit command](#) if the editor you use is different from the system editor. Click [Apply](#).
4. Complete the GUI Page. Deselect [Multiple object windows](#) if you will use only one work area at a time. Click [Apply](#).
5. Complete the Extract Page. Specify where you want extracted TeamConnection data to be stored on your workstation or LAN. Deselect [Read-only](#) if you want to be able to edit extracted data. Click [Apply](#).
6. Complete the Pool Page. You can [enter the build pool name](#) now or wait until you are ready to build the parts that make up an application. Your Build Administrator will give you the pool names for your project. Click [Apply](#). Click [OK](#) to close the Settings Notebook window.

Note: The GUI defaults you have set do not apply to commands you type yourself, either from the operating system prompt or the command line provided by the GUI. To set defaults for issuing commands, you must edit your config.sys file.

* * * * *

X Card 22000 (Procedure2 Task2)

Edit CONFIG.SYS

At installation, your CONFIG.SYS file was updated to include default values for issuing TeamConnection commands. You can modify or add defaults to make issuing commands easier.

To edit the CONFIG.SYS file:

1. Type 'e config.sys' at the operating system prompt.
2. Scroll down to the section containing commands beginning with SET TC_.
3. Add or modify the following lines as desired to set environment variable defaults:

```
SET TC_FAMILY=[family name]
SET TC_COMPONENT=[component name]
SET TC_RELEASE=[release name]
SET TC_USER=user name to logon]
SET TC_BECOMEUSER=[user name when working]
SET TC_BUILDPOOL=[build pool name]
SET TC_WORKAREA=[work area you use]
```

Note: The variables you choose as defaults depend on the specifics of your working environment. For example, if you work on various releases, do not set COMPONENT and RELEASE variables.

You can override any set value by including the corresponding flag in the command you want to issue. Refer to *TeamConnection Commands Reference* for details on setting variables and issuing commands.

VisualAge Generator uses environmental variables for application generation and preparation in the TeamConnection environment. These variables set default builder names, so if you change the name of a default builder you will have to change it in your CONFIG.SYS file.

[Note the Intervention Engine mark-up for this card.](#)

* * * * *

Skipping now to the end of the *cue card set*
Note that there are over 140 cards in the VisualAge Generator help module.

X Card 140000 (Procedure9)

Run a Query / Report

To view the data you need from the TeamConnection database, you use the GUI to form queries that issue the commands that TeamConnection understands.

VisualAge Generator uses a Library Search Mask Specification window to filter names of parts existing in TeamConnection associated with the current Library MSL.

[Double-click a task to read about that topic.](#)

Use the Filter Window
Complete a Query
Select Appropriate View
Save View as Default
Save Query as Default
Add Query to Task List
Save Report as a File
Print a Query Report
View the Change History
Use the Library Search Mask Specification Window

* * * * *

Card 141000 (Procedure9 Subtask1)

Use the Filter Window

Filter windows allow you to fill in search criteria that the GUI converts into a query. Queries can be broad (Show all) or specific (like% or an exact match you type). The windows display search attributes to complete and provide a [History](#) field to show previously defined queries. The [Query](#) field displays the actual query command that is the result of your selections.

There are three Filter windows you can use to find parts within TeamConnection:

- [Parts](#). Limits search to a specific context.
- [PartFull](#). Searches across releases, components and work areas.
- [BuildView](#). Searches for build related information.

Working with part filter windows:

1. From the Task window select [Objects: \(choice\) -> \(choice\)](#). If the choice is not Parts, a filter window

will display with the name of the choice.

If you selected [Objects: Parts ->Parts](#), the Parts filter window will display. Selecting [Objects: Parts -> PartFull](#), displays that filter.

2. From filters other than Parts you can click [Show all](#), to display data from which you can select the specific parts you want. If you choose components or releases and then double-click [Show: Parts](#), the PartFull window will display with all the parts across these releases or components.

* * * * *

The following [text](#) example for naming conventions in VisualAge Generator, as applied to TeamConnection, indicates the highly technical nature of the product throughout as it interfaces with code generation for deployment on various platforms.

[Card 23200 \(Generator Procedures2Task5\)](#)

Overview of TeamConnection Naming Conventions

Using VisualAge Generator with TeamConnection requires adhering to certain naming conventions.

1. VisualAge members are identified to the TeamConnection information model by the use of delimiters as follows:
empty prefix delimiters at the beginning of the member name [< >](#) and;
suffix delimiters containing a colon [< : >](#) at the end of the name.
You see these delimiters when VisualAge Generator members have been migrated to TeamConnection and you must use them when naming these parts for any actions from the TeamConnection GUI.
2. Further identification requires part extensions and part type assignments. For example, a file with the extension of [.APP](#) has a part type assignment of [fdcVGApplication](#) that you must use whenever part type is required. Refer to the VisualAge Generator [Help](#) file for extensions and part type assignment details.

The TeamConnection information model is designed to provide common object definitions for information used by various tools supporting application development. The data gathered in most VisualAge Generator member types is not common to other tools, but the [global data item member type](#) is the exception. It defines data definitions that could be shared with tools that provide functions such as data modeling and database definition. To provide a mechanism for these tools to share definitions, TeamConnection defines an object called a [DataElement](#). When you create a global data item, TeamConnection creates two objects: a global data item with a part name of [<> MEMBERNAME.DTE <:>](#) whose part type is [fdcVGDataItem](#) and;
a Data Element with a part name of [<> MEMBERNAME <:>](#) whose part type is [fdcDataElement](#).

3. TeamConnection parts can be named using both upper and lower case characters. But, because VisualAge Generator uses upper case only for member names, it will be easier to remember new names if you always use upper case characters when creating parts.
4. TeamConnection builds map groups NOT individual map members, so naming map members from VisualAge Generator requires the use of the full path name:
MAPGROUP.MAPNAME.MAP.
 For all other members, use the member name as shown in the Member List and a part type extension: **MEMBERNAME.EXT**
5. Input generation control files created from the TeamConnection GUI require the application name that is being generated and the part type extension. Drive characters cannot be included but relative paths can be used to indicate the same type of file for separate environments. For example:
 the Link edit control generation option files might be named **MVS\LEARN.LED** and;
VSE\LEARN.LED.
6. Output placeholders created from the TeamConnection GUI can be any name you choose. For example **TESTFAM.CLIENTS** might be a collector placeholder for the GUI code on your workstation. The part type is *file*, though upon creation you selected None, since it had no content then.
7. Override options and "results output" files that are automatically placed in your Library MSL and TeamConnection work area as a result of the initiation of a member build, follow this format: **SYSTEM\MEMBERNAME.EXT**.

system is the target system;
membername is the member generated or prepared;
ext is the type of output;
 the part type is *file*.

For example, the files created on an initial build tree in an OS/2 client/server environment for a member named **LEARN** would be:

OS2GUI\LEARN.EZEGEN - a generation results collector file;
OS2GUI\LEARN.EZEGENOUT - a generation output file;
OS2GUI\LEARN.OVR - an override options file;
EZEPREP AND EZEPREPOUT - preparation output files.

Preparation is not required for GUI applications, but for the sake of consistency across member-parts, these output files are always created.

* * * * *

The Intervention Engine

*The Cue Card **Intervention Engine** for OS/2 was built by Framework, Inc. in Tarrytown NY for IBM. It is one of many Electronic Performance Support Systems (EPSS), called "interventions," that can be integrated into applications to provide a step-by-step level of support. The cue cards float on top of the application so that users don't have to hop back and forth to refer to the support materials.*

As Lead Writer, I often coded and compiled text myself to observe a new card's appearance, rather than wait to view it after its compilation in a new "release." Below is a brief example of the engine conventions as marked-up for the Cue Card 22000 "[Edit CONFIG.SYS](#)," given as an example in its text format earlier.

```
.cuecard 22000 template 503000
:attributes.
:eattributes.
:objects.
:texregion 1.
    :font "Helvetica" 10. :color "darkblue".
    "Edit CONFIG.SYS".
:etextregion.
:texregion 2. :font "Helv"9.
    "At installation your config.sys file was updated to include default values".
    "for issuing TeamConnection commands. You can modify or add defaults to".
    "make issuing commands easier.".
:nl. :nl.
"To edit the config.sys file:".
:nl.
:olist.
:li. "Type". :color "blue". "e config.sys". :color "black". "at an OS/2 prompt.".
:li. "Scroll down to the section containing commands beginning with SET TC_.".
:li. "Add or modify the following lines as desired to set environment variable defaults:".
:eolist.
:nl :nl.
:color "blue".
    "SET TC_FAMILY= [family name]".
:nl.
    "SET TC_COMPONENT= [component name]".
:nl.
    "SET TC_RELEASE= [release name]".
:nl.
    "SET TC_USER= [user name to logon]".
:nl.
```

```
"SET TC_BECOMEUSER= [user name when working]".
:nl.
"SET TC_BUILDPOOL= [buildpool name]".
:nl.
"SET TC_WORKAREA= [work area you use]".
:nl. :nl.
:color "black".
>Note: The variables you choose as defaults depend on the specifics of your working
environment.".
"For example, if you work on various releases, do not set COMPONENT and
RELEASE ".
"variables.".
:nl :nl.
    "You can override any set value by including the corresponding flag in".
    "the command you want to issue. Refer to".
    :italic. "TeamConnection Commands Reference". :eitalic.
    :for details on setting variables and issuing commands.".
:etexregion.
:eobjects.
:ecuecard.
```
